

UNIVERSITY OF TWENTE

DEPARTMENT APPLIED MATHEMATICS

CHAIR HYBRID SYSTEMS

---

**Robust sensor coverage in a  
two-dimensional area**

---

*Candidate:*  
Floris BROEKEMA

*Supervisor:*  
P. FRASCA

April 4, 2015

# Preface

This report describes a master thesis about robust sensor coverage. The assignment is done within the University of Twente at the chair hybrid systems, of the department mathematics. The time assigned to this thesis is about 32 weeks, which is equal to 40 European credits. The supervisor of this master thesis is Paolo Frasca, working at the chair hybrid systems. Among other subjects he has done research in the fields of hybrid systems, system control, opinion dynamics and robust sensor coverage in a 1-dimensional space.

Within this master thesis, the task is to find a good or even optimal deployment of sensors in a 2-dimensional space. Besides that, the sensors can fail with a certain probability which means that the solution must be robust, such that it takes the possible failures into account. Various properties are known for the 1-dimensional space, the intention is to extent this to a 2-dimensional space.

For this master thesis I would like to thank Paolo Frasca for his guidance, time and work throughout the whole research. When I was stuck at some point in the research he always helped me out and always gave very good advice. Furthermore I would like to thank the department of mathematics as they helped me through my study which is now concluded with this master thesis.

# Abstract

This master thesis is about the quality of sensor networks in a 2-dimensional space. The sensors are assumed to fail independently with a certain probability. In former research, boundaries for the cost of sensor networks for the 1-dimensional space, relations for coverage conditions and algorithms for sensor deployments without failures are stated. In this thesis the average cost of a sensor deployment will be analyzed with the help of Voronoi diagrams and properties of the cost function are derived. Two common used deployments, the grid deployment and the random uniform deployment are defined and simulated. Also it will be shown that the grid deployment is not optimal, even when the sensors cannot fail. Furthermore, it will be shown that the cost of the uniform deployment is higher than the cost of the grid deployment and that these costs seem to be parallel to each other, both functions logarithmic with respect to the number of sensors. A general lower bound for the cost of sensor deployments will be constructed which holds for all sorts of sensor deployments and from the coverage conditions of the random uniform deployment a lower bound for the cost is constructed for large numbers of sensors. This lower bound is in most cases more strict, but only holds for random uniform deployments. To find the best deployment which minimizes the cost, two discrete algorithms are made which aim to optimize this goal. The first one is a gradient descent algorithm and the second is derived from the first but has faster convergence. For the gradient descent algorithm, it is proved that it converges to critical points which are local optimal sensor deployments. Then in simulations it is shown that the algorithm performs better than the grid and uniform deployments. For the second algorithm the convergence properties will not be analyzed in detail as the algorithm is very similar to the first algorithm.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1	Problem description . . . . .	4
2	Former research . . . . .	5
<b>2</b>	<b>Model</b>	<b>7</b>
1	Mathematical problem description . . . . .	7
2	Geometrical properties of the cost function . . . . .	8
<b>3</b>	<b>Grid and random uniform deployment</b>	<b>11</b>
1	Defining the deployment strategies . . . . .	11
2	Non optimality of the grid . . . . .	12
3	Simulation results . . . . .	13
<b>4</b>	<b>Cost analysis</b>	<b>17</b>
1	A general lower bound . . . . .	17
2	Known theorems about sensor coverage . . . . .	18
3	Comparison with simulations . . . . .	21
<b>5</b>	<b>Sensors deployment algorithms</b>	<b>24</b>
1	Gradient descent algorithm . . . . .	24
2	Gradient based algorithm . . . . .	31
<b>6</b>	<b>Discussion</b>	<b>34</b>
<b>7</b>	<b>Conclusion</b>	<b>36</b>
<b>8</b>	<b>Recommendations</b>	<b>37</b>

# Chapter 1

## Introduction

### 1 Problem description

This master thesis is about robust sensor coverage in a two-dimensional space. Sensor coverage means that an area (in this case a two-dimensional area) must be filled with sensors such that the sensors can get information from the whole area. Possibly, the sensors only have a certain sensing radius which can be seen as a disk around the sensor in which the sensor can measure. Sensor coverage is important for many systems, for example a smoke detection system. In a smoke detection system the whole building must be covered with sensors such that smoke is detected as fast as possible. In researches on sensor coverage, there are two different aims, namely to use the least amount of sensors to cover the whole area, or to use a fixed number of sensors to cover the area as good as possible. In this research the aim is to maximize the quality of the sensor network and to find the best sensor covering with a fixed number of sensors.

The quality of a sensor deployment can be measured in various ways. Central in these measures is the distance from a point in the area to the closest sensor. In this report the euclidean distance will be used to determine the distance from the sensors to any point in the area. The quality of a sensor network will be defined as the (weighted) average distance of each point of the area to the closest sensor. However, in this research the quality of a sensor network is defined as the distance to the worst detectable point. Throughout this report, the quality of a sensor deployment is called the cost of the sensor deployment. In other words, the cost is the maximal distance from any point in the area to the closest sensor. This measure relies on the worst case scenario. To restrict the problem and to avoid assumptions on the cost definition, only a convex area will be considered, namely the unit square area. Most findings can be adapted to other convex areas but this will not be done within this research.

Besides finding a good sensor deployment which covers the unit square area as good as possible, robustness is very important. Because the quality is defined as the worst case performance and thereby the robustness will be obtained when maximizing the quality. However, each sensor can be active or inactive. An inactive sensor is also called a sleeping or failing sensor and cannot detect anything while it is sleeping. Therefore the cost only depends on the active sensors. In this research, the sleeping sensors are determined probabilistically with identical and independent probabilities. When deploying the sensors in the area the probability on failure is assumed to be known and the possible failures should be taken into account. Because possible sleeping sensors are taken into account, the robustness increases. With the cost, the performance of a sensor deployment can be rated and thereby a certain performance can be guaranteed.

In chapter 2 of this research the general problem statement and details about the cost computation will be given. Properties of the cost function will be derived which are important to understand the behavior of the cost function and these will be used later on in this report. In chapter 3, two deployment strategies, namely the random uniform and the grid deployment, will be defined and discussed. Cost simulations will be done and the strategies will be compared. Besides that, the two deployment strategies will be used throughout the report for theorems and simulations. Furthermore, known relations between the cost of a sensor deployment, the number of sensors and the probability on active sensors will be analyzed and a general lower bound will be constructed in chapter 4. With this lower bound, a minimal cost can be deduced for all sensor deployments. This lower bound will be compared to known asymptotic relations on the sensing radius which only hold when the number of sensors goes to infinity. In chapter 5 two deployment algorithms will be defined which aim to find optimal sensor locations which minimize the cost. One of these algorithms will be the gradient descent algorithm and the other is derived from the gradient descent algorithm. Simulations are done for both the algorithms and the critical points and the convergence will be analyzed. Then the results of the whole research will be concluded and discussed. At last recommendations for further study are presented.

## 2 Former research

Much research has been done on sensor coverage problems. In this research the sensors are assumed to have fixed locations, but in other researches the sensors can be dynamic and have to move through an area. Some examples of studies that are related to this study are: a book which discusses most sensor coverage problems in general [1]; papers on algorithms for making good sensor deployments [2][3]; papers on dynamical coverage problems with restricted communication radius [4] and papers on analyzes of the asymptotic relations between the number of sensors and the minimal radius needed for coverage [5][6]. Below a short description of several researches are given and is described how they are used in this research.

The problem of robust sensor coverage has also been studied in the one-dimensional setting [7]. This research considers the same kind of assumptions such as stationary sensors, the probabilistic sensor failures and the same cost function. The sole difference is the dimension in which the sensors are deployed. consequently, the same methods are considered and if possible extended to the two-dimensional case. In one dimension, the problem can be described as a linear program and then be solved. Furthermore, in this research ([7]) an equispaced sensor deployment is defined and analyzed for its cost and relative cost to the optimal solution. Besides that, the asymptotic relations are examined together with the performance of a random uniform deployment.

Explicit asymptotic relation can be found between the number of sensors [6], the probability on sensor failures, the sensing radius and the probability that the area is covered. In this research we want to have an explicit relation for the sensors, the probability on sensor failures and the cost. The cost can be seen as the minimum radius required to cover the area with a certain sensor deployment. So the relation stated in the paper can be used and is studied for our own research. For finding a lower for the cost, the relations from the paper are used.

Besides these papers about analytic approaches to provide information about the cost, researches have been done on algorithms to deploy sensor networks, both on stationary or dynamic sensors. In these researches Voronoi partitions are often used in algorithms to find good sensor deployments. However, sensor failures are not taken into account in these algorithms. A Voronoi partition is an area around a sensor with all the points which are closest to that sensor. In one of the researches [3] an algorithm is defined and analyzed in a problem setting which is very similar to the two-dimensional case sensor coverage problem studied in this research. A big difference is that sensors cannot fail. This results in a different cost function. Therefore a new algorithm will be developed and analyzed which is based on the gradient descent algorithm.

# Chapter 2

## Model

In this chapter the mathematical problem description will be defined and explained. First the variables will be defined such that a cost function can be made. Then the cost function will be explained and described with more detail. This is done to improve understanding of how the cost behaves for different sensor deployments.

### 1 Mathematical problem description

In this section the general problem will be described with a mathematical model. Let  $Q$  be a convex two-dimensional space on which sensors are deployed. For simplicity, let  $Q$  be the square unit surface,  $[0,1] \times [0,1]$ . On this area  $n > 0$  sensors must be deployed. Let the locations of the sensors be  $x_i$ ,  $i = 1,2,\dots,n$ . The sensors must be deployed within  $Q$ , so  $x_i \in Q$ . Each sensor is active with probability  $p$  and not active with probability  $\bar{p} = 1 - p$ . The failure events of different sensors are assumed to be independent and identically distributed.

Now we can define  $C(x_A)$  as the cost of the sensor deployment  $x$  where  $A$  is the set active sensors. This cost is given by the worst case performance which is equal to the distance of the farthest point in  $Q$  to the closest active sensor. For the distance the 2-norm will be used to calculate the Euclidean distance. Then the cost of an active set sensors is equal to

$$C(x_A) = \max_{s \in Q} \min_{i \in A} \|s - x_i\|_2. \quad (2.1)$$

To be complete, the cost with no active sensors is defined as the largest distance between two points in  $Q$ , which is  $\sqrt{2}$ . The set active sensors  $A$  can be any subset of  $\{1,2,\dots,n\}$ . Now let  $E_A$  be the event that that the set active sensors is equal to  $A$ . Then the probability that event  $E_A$  occurs can be calculated as:

$$Pr(E_A) = p^{|A|}(1-p)^{n-|A|}. \quad (2.2)$$

Now the cost function for the whole sensor deployment can be made where all possible active subsets due to sensor failures are taken into account. This can be done by a weighted sum over all the costs for the different active sets  $A$ , multiplied by the probability on event  $E_A$ . This way the total cost function is a weighted average over all events, namely the expected value of the cost of a sensor deployment. With this definition of the cost of a sensor deployment  $x$ , the cost can be computed as follows:

$$C(x) = \mathbb{E}(C(x)) = \sum_{A \subseteq [n]} Pr(E_A)C(x_A). \quad (2.3)$$

Throughout the whole research this is the measure of quality for a sensor deployment. Therefore, finding the sensor locations  $x$  which minimize this cost function for a given  $p$  is the main goal of this research. With use of this cost function an objective function can be constructed. Because the cost with no active sensors is not different for different deployments, it is not part of the objective function:

$$\min_{x \in Q^n} \sum_{A \neq \emptyset, A \subseteq [n]} Pr(E_A) C(x_A) \quad (2.4)$$

## 2 Geometrical properties of the cost function

As described in the previous section, a cost function describes the total cost of a sensor deployment with stochastic failures of sensors. In this section the cost of an set active sensors will be described with more detail. Besides that, a method to calculate these cost will be explained. As defined in the previous section, the cost of a set active sensors is

$$C(x_A) = \max_{s \in Q} \min_{i \in A} \|s - x_i\|_2. \quad (2.5)$$

This formula describes how to calculate the distance from the farthest point to the closest sensor. With help of Voronoi regions ( $V_i$ ) this formula can be rewritten. But first will be explained what Voronoi partitions are. Each active sensor has its own Voronoi partition which contains the points of  $Q$  which are the closest to that sensor. So a Voronoi region of a sensor depends on the neighboring sensors and is defined as

$$V_i = \begin{cases} \{s \in Q \mid \|s - x_i\|_2 \leq \|s - x_j\|_2, \forall j \in A\} & \forall i \in A, \\ \emptyset & \forall i \notin A. \end{cases} \quad (2.6)$$

In figure 2.1 a Voronoi diagram can be seen for a random deployment with 9 active sensors. As can be seen, the cost is attained at only one point of  $Q$ . The cost is defined by the point which is the farthest away from the closest sensor. This is the same as calculating the farthest point in each Voronoi region  $i$  to the corresponding sensor  $i$  and then maximizing over all active sensors. This because the points in the Voronoi partition of sensor  $i$  solely contains the points which are the closest to sensor  $i$ . Now the cost function can be rewritten as:

$$C(x_A) = \max_{i \in A} \max_{s \in V_i} \|s - x_i\|_2. \quad (2.7)$$

This formula can even be made more precise. From figure 2.1 it is clear that the points in a Voronoi partition which are the farthest away from the corresponding sensor, are the points on the boundaries of the Voronoi regions. Let  $\delta V_i$  be the edge of the Voronoi partition, then the formula can be rewritten as:

$$C(x_A) = \max_{i \in A} \max_{s \in \delta V_i} \|s - x_i\|_2 \quad (2.8)$$

However, not all points on the edge of a Voronoi partition are points on which the maximum distance to the closest sensor can be obtained. Let  $W$  be the set of points at which the cost is attained:

$$W = \left\{ a \mid \max_{i \in A} \max_{s \in \delta V_i} \|s - x_i\|_2 = \|a - x_i\|_2, \forall s \in Q \right\} \quad (2.9)$$

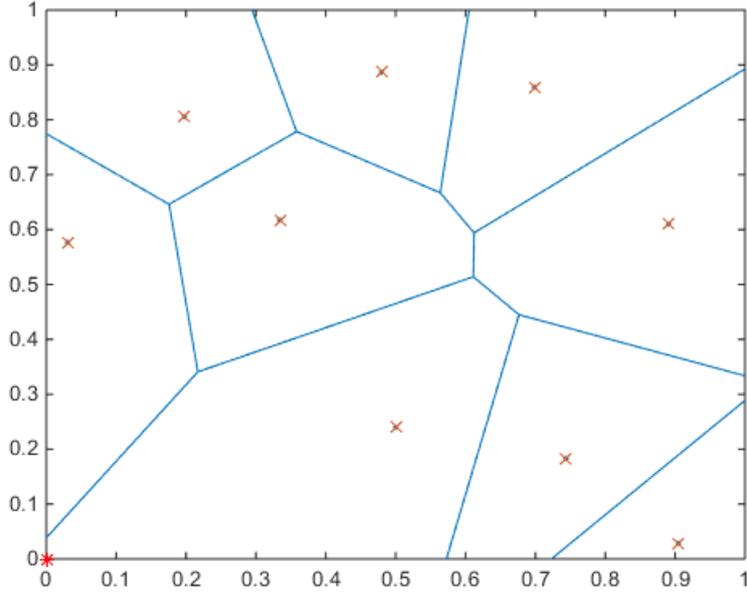


Figure 2.1: A figure of Voronoi partitions created in MATLAB for a random deployment in the unit square surface. The red crosses are the sensors and the blue lines are the boundaries of the Voronoi partitions. The cost is attained at the red star at (0,0) and is equal to 0.5546.

**Proposition 2.1.** *The set  $W$  can be described as a subset of the union of 3 types of points, namely the vertices of the Voronoi diagram ( $W_1$ ), the points at which the edges of the Voronoi partitions intersect with the edges of  $Q$  ( $W_2$ ) and the vertices of  $Q$  ( $W_3$ ).*

$$W \subseteq W_1 \cup W_2 \cup W_3 \quad (2.10)$$

Moreover, these 3 types of points determine the cost of a sensor deployment.

*Proof.* First will be explained why the maximal distance to the closest sensor can be obtained by the points of  $W_2$  and  $W_3$ . Consider a sensor with an edge of  $Q$  in its Voronoi partition. Then there is a point on the edge of  $Q$  in the Voronoi diagram which has the smallest distance to the corresponding sensor. This point is obtained by moving in a straight horizontal or vertical direction until the edge of  $Q$  is reached. From this point, moving along the edge of  $Q$  will increase the distance to the sensor (Pythagoras) and this can be done until the Voronoi diagram will be exceeded. The last point before exceeding the Voronoi partition has to be either a corner of  $Q$  or an intersection of the Voronoi boundary with an edge of  $Q$ . So this point is in  $W_2$  or  $W_3$  and might have the maximal distance to the closest sensor, namely the sensor corresponding with that Voronoi partition.

For the vertices of the Voronoi diagram ( $W_1$ ) a similar argument can be used. Consider two neighboring sensors  $i$  and  $j$ , then for the edge of the Voronoi diagram in between the two sensors we know that the distances to the sensors are equal. Also 1 point, namely the point between the 2 sensors has the smallest distance to the sensors and lies on the edge of the Voronoi partition. Moving along the Voronoi edge from this point will increase the distance to the sensors. This can be done till the end of the Voronoi partition, this point will be either a vertex of the Voronoi diagram or an edge of the area  $Q$ . Here the point on the edge of the area  $Q$  is again a point from  $W_2$  and the vertex of the Voronoi partition is in  $W_1$ .

So within a Voronoi partition the maximal distance to the corresponding sensor is attained at one of these three types of points. Now by (2.7), maximizing over all Voronoi partition will result in the cost. So the cost depends on the vertices of the Voronoi diagram, the intersections of the

edges of  $Q$  and the Voronoi partitions and the vertices of  $Q$ . This means that  $W$  is a subset of the union of these points.  $\square$

**Remark 2.1** (One-dimensional linear program). *In the one-dimensional case it is known that the problem can be written as a linear program. Then the cost is depending on the distances between the sensors and between the begin/end of the interval and the first/last sensor. These relations are linear and thus can be used to write a linear program. In the two-dimensional problem the distances are calculated by the 2-norm which is not linear. Besides that, the points of  $W_1, W_2$  and  $W_3$  which determine the cost as described in proposition 2.1 cannot be determined with linear equations either.*

**Remark 2.2** (Relation to disk-covering problem). *When the probability  $p$  on active sensors is equal to 1, the cost function as a weighted average over all possible events is reduced to a cost function with only one event. Only the event where all sensors are active is left. Using Voronoi partitions, the objective function can be written as:*

$$C(x) = \max_{i=1,2,\dots,n} \max_{s \in \delta V_i} \|s - x_i\|_2 \quad (2.11)$$

*This function must be minimized to find an optimal deployment in which the sensors cannot fail. This problem is very similar to the disk-covering problem where an area must be covered with circles with a certain radius. The difference is that in the disk-covering problems they increase the number of sensors and in this research, the radius is increased to cover the whole area. However, the solutions are very similar and it can be shown that the optimal solution of the objective function without failure rates is a deployment where the locations of the sensors are the centers of circumcircles of the Voronoi partitions[3]. In other words, circles can be drawn with the sensors as center which passes through all the vertices of the Voronoi partitions.*

Now only 3 types of points have to be considered when calculating the cost of a sensor deployment. The cost is defined as the sum over all possible sets active sensors, therefore the number of terms in the sum raises exponentially with respect to the number of sensors. As a result the needed computational time is growing exponential too. To avoid exploding computation time, stochastic simulation will be done such that the number of sensors can be increased. The cost function (2.3) is a weighted sum over all possible events weighted with the probabilities that the events occur. That can be sampled by drawing the set of active sensors from a binomial distribution with chance  $p$  on success. Then the costs can be calculated by computing the distance of each of the points in  $W_1, W_2$  and  $W_3$  to the closest sensor. Then the maximal distance from any of these points to the closest sensor is the cost for the sensor deployment with the corresponding set active sensors. The exact cost can be estimated by repeating the sampling and cost calculation and then taking the average of these outcomes. Later in this report will be defined how many time this will be repeated in order to obtain a good estimation of the cost.

## Chapter 3

# Grid and random uniform deployment

In this section the grid and the random uniform deployment are described. These two deployments will be used to compare the algorithm which will be described in a later section. Furthermore some properties will be described with respect to the cost and lastly some simulation results will be shown.

### 1 Defining the deployment strategies

For the one-dimensional case the cost are known [7] for the equispaced sensor deployment. For failure rates close to 0 this deployment has (near) optimal cost. A similar method will be used in the two-dimensional case, here the equispaced deployment becomes a grid. The equispaced sensor deployment is defined as  $x = \frac{1}{2n}(1, 3, \dots, 2n - 1)$ , using this same structure, the grid with  $n$  sensors is defined as:

$$x = \left( \frac{1}{2\sqrt{n}}(1, 3, \dots, 2\sqrt{n} - 1), \frac{1}{2\sqrt{n}}(1, 3, \dots, 2\sqrt{n} - 1) \right) \quad (3.1)$$

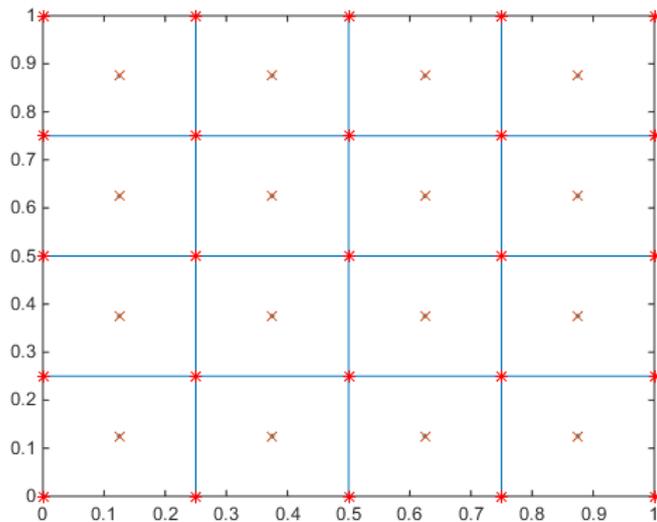


Figure 3.1: A figure of a grid sensor deployment with 16 sensors with Voronoi partitions created in MATLAB. The red crosses are the points in  $W_1$ ,  $W_2$  or  $W_3$  and they all belong to  $W$

The grid is only defined with a square number of sensors and has a gap of  $\frac{1}{2\sqrt{n}}$  between each two sensors and between the boundary of  $Q$  and the closest sensors. Therefore, the cost of the sensor network is equal to  $\sqrt{\frac{1}{2n}}$  when all sensors are active.

Another deployment strategy is the random uniform deployment, in this deployment all the sensor locations are drawn from a uniform distribution on  $[0,1] \times [0,1]$ . Because the locations are drawn from a random uniform distribution, the locations of the sensors and thus the cost of the deployment are unknown and can always be different. As can be seen in fig 3.2 the maximal distance is only obtained once in most cases with random uniform deployments. Therefore the cost is more dependent on that sensor which has this maximal distance within its Voronoi partition. When this sensor becomes inactive, another sensor will attain the maximal distance to the farthest and the cost will increase. For other sensors which are relative close to each other the opposite holds. When one of these sensors becomes inactive, the cost of the active sensor network may remain the same.

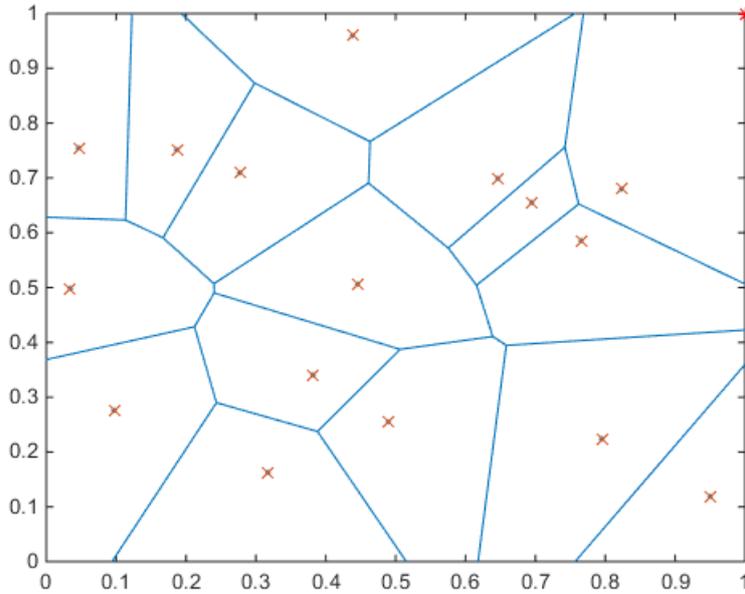


Figure 3.2: A figure of a random uniform sensor deployment with 16 sensors with Voronoi partitions created in MATLAB. The red cross is the sole point of  $W$  which attains the cost as distance to the closest sensor and belongs to  $W_3$ .

## 2 Non optimality of the grid

As stated, the equispaced sensor deployment in the one-dimensional setting has (near) optimal cost for values of  $p$  close to 1 and has optimal cost for  $p = 1$ . That is, when all sensors are active the cost of the equispaced sensor network is minimal. The question which arises, is whether this is also true for the two-dimensional grid deployment in the two-dimensional setting. However, the optimal sensor locations are not known and the optimal cost therefore is not known, it can be shown that the cost of the grid deployment with no failures is not minimal when 9 or more sensors are deployed. This can be explained. In the grid deployment all sensors have 4 points with maximal distance to the sensor. When a circle is drawn through these points the actual range the sensor could cover with the same radius has much overlap with the neighboring sensors and that radius is not fully used.

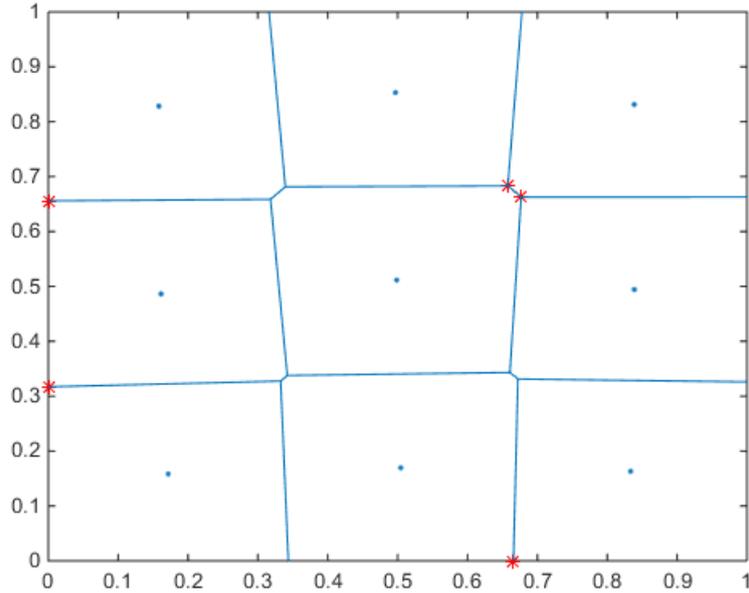


Figure 3.3: A sensor deployment with a lower cost than the grid deployment for  $p = 1$ . The red crosses are the points in the Voronoi regions which are the farthest away from the closest sensor. Since only 5 crosses can be seen, there must be some overlap.

In figure 3.3 a sensor deployment with a lower cost than a grid deployment can be seen. The cost of the grid and the improved deployment with 9 sensors,  $p = 1$  are 0.2357 and 0.2341. However the difference between the costs is very small, it is shown that the performance of the grid is not optimal with  $p = 1$ . When the probability on active sensors decreases the grid is performing relatively worse. For the uniform deployment less is known about the cost. The sensors are drawn from a uniform distribution over  $[0,1] \times [0,1]$  and therefore the expected location of the sensor is in the middle. The expectation of the cost however is not as simple as that and depends on the location of other sensors, as well on the probability  $p$  on active sensors. Though it may be the case that the random uniform deployment is better than the grid deployment for small  $p$ .

### 3 Simulation results

Now some simulation results will be described and examined. The simulations are done as described in chapter 2 and the cost is estimated by an average of the costs of all simulation runs. The cost in a simulation run is the maximal distance from each of the distances from any points in  $W_1$ ,  $W_2$  and  $W_3$  to the closest sensor. Each run the grid deployment is constructed and a uniform random deployment is sampled according to uniform distribution over  $[0,1] \times [0,1]$ . So the random uniform sensor deployment may be different for each cost calculation. Then the sleeping sensors are drawn from a binomial distribution with probability  $p$  on an active sensor. This is done once for each run so both the deployments have the same number of active sensors. Then the costs are calculated as described for the active set sensors for both the random uniform and grid deployment. This will be repeated 100 times and the average of these values is the estimator for the costs. This represents the cost function very well because the cost function is a weighted sum over all possibilities. So the expected value of the estimator should equal the cost for a large number of runs.

In figure 3.4 the estimated costs of the random uniform and the grid deployment with a probability on active sensors equal to  $p = 0.9$  are shown with logarithmic scales. As expected the grid performs better than the random uniform deployment for large  $p$ . In the plot with the logarithmic scales can be seen that the estimated costs of the grid deployment and random uniform deployment seem to approximate two straight lines which are parallel to each other. Therefore, a logarithmic relation between the number of sensors and the cost of both the deployments can be expected. From figure 3.4 it is also expected that the asymptotic cost ( $n \rightarrow \infty$ ) is not equal for the grid and random uniform deployment with  $p = 0.9$

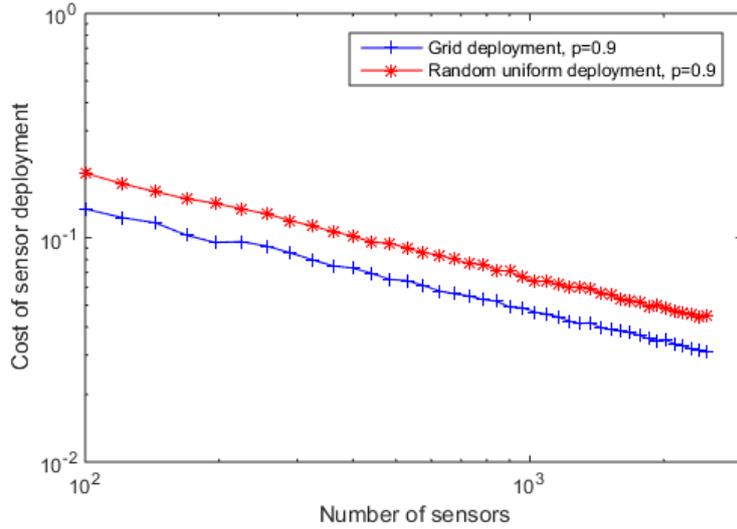


Figure 3.4: This graph shows the estimated costs of the grid deployment ( $p = 0.9$ ) and the random uniform deployment ( $p = 0.9$ ) for square numbers of sensors. Both the axis are logarithmic.

In figure 3.5 can be seen how the cost behaves when the probability on active sensors increases from 0 to 1 in 50 steps. With  $p \approx 0$  the difference between the cost of the two strategies is small and the cost is decreasing when  $p$  increases. So the grid always seems to have similar or better results regardless of the value of  $p$ . Therefore, it can be concluded that the grid is a better strategy than the random uniform deployment. However, as already stated in this report, the grid deployment is not the optimal deployment, even for  $p$  equal to 1.

In both the simulations each setting with a different probability on active sensors  $p$  or a different number of sensors  $n$  is repeated 100 times. Then an average is taken over all these repetitions, but then it is important to know whether 100 repetitions is enough or whether the variance is rather high. Therefore the variance of the simulations is discussed here. Because the probability distribution of the cost is unknown, the sample variance ( $s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$ ) will be used to determine the variance of the simulations. Then the sample variance is divided by the average of the runs to obtain the ratio of variance with respect to the average. In figure 3.6 can be seen how the relative variance decreases when the number of sensors is kept equal and the probability on active sensors increases. Besides that, the ratio variance with respect to the average of the grid deployment is in most cases greater than that of the random uniform deployment, this can be explained because the random uniform deployment is already random and may differ each repetition. With both the deployments, the variance is considered very low, thus no more repetitions are required.

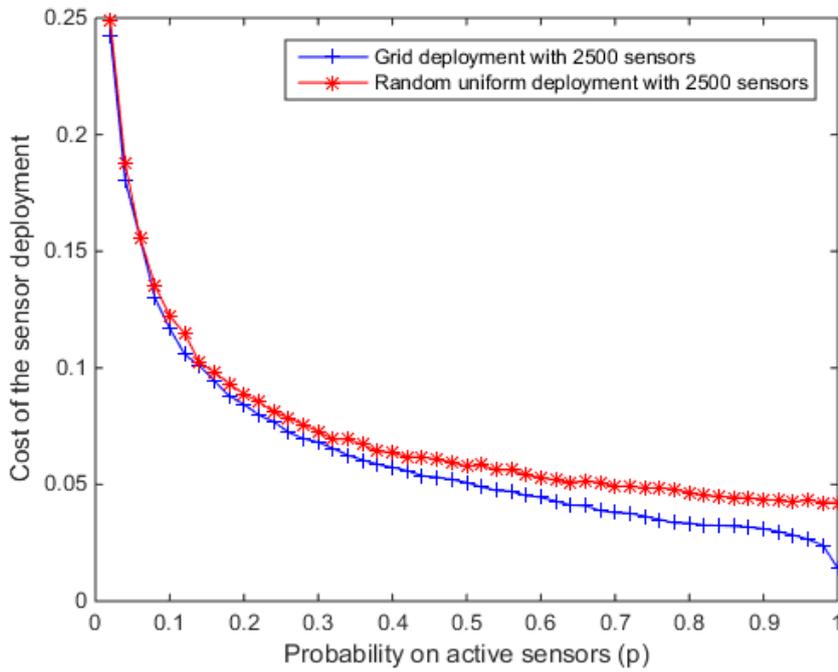


Figure 3.5: The costs of the grid and random uniform sensor deployments for different values  $p$ .

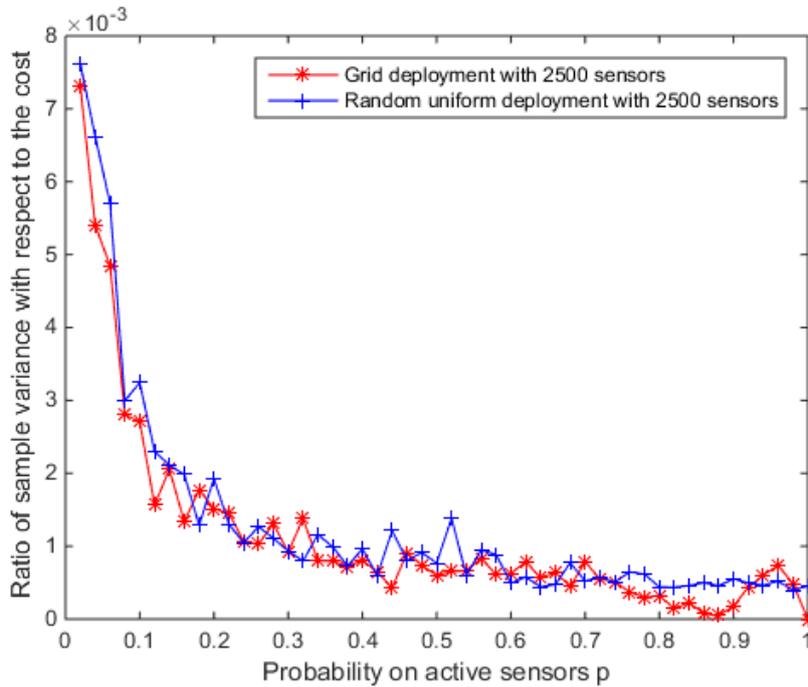


Figure 3.6: The variance of the cost of a grid and random uniform deployment divided by the average of the repetitions. The number of sensors stays equal at 2500 and the probability on active sensors is altering from 0 to 1.

When  $p$  is fixed and the number of sensors increases a similar result can be seen in figure (3.7) where the ratio is decreasing as the number of sensors is increasing. This can be explained by looking at a grid deployment. The grid deployment shows much symmetry and thereby the cost is often attained at more than one point and by more than one sensor. This means that there are various different active sets of sensors with the same cost and thus the sensor deployment is less affected by the sleeping sensors which can be at different locations. When the number of sensors increases, the number of different active sets sensors with the same cost increases, therefore the variance is relatively smaller for larger number of sensors.

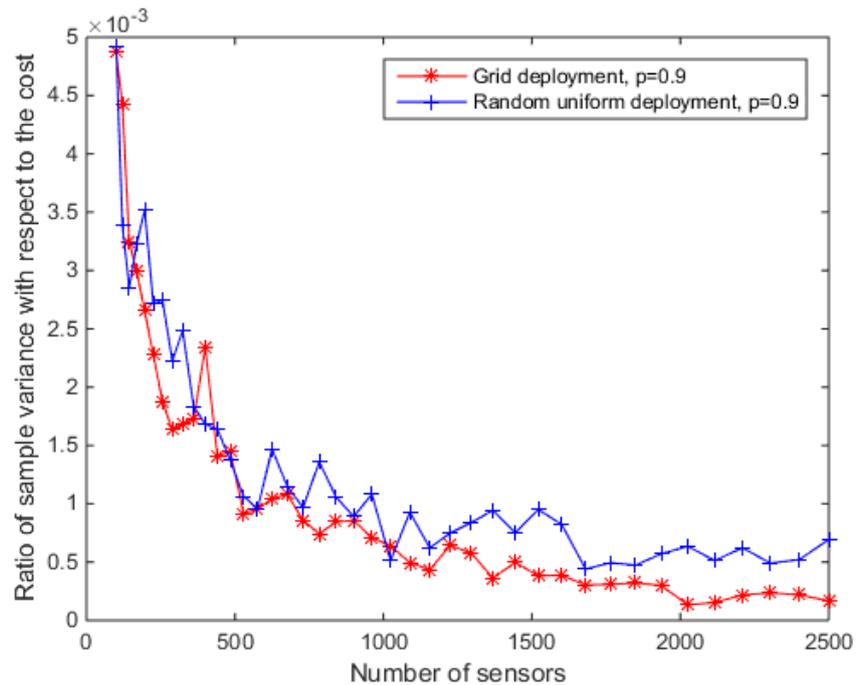


Figure 3.7: The variance of the cost of a grid and random uniform deployment divided by the average of the repetitions. The number of sensors stays equal at 2500 and the probability on active sensors is altering from 0 to 1.

# Chapter 4

## Cost analysis

In this chapter the cost and behavior of the cost is examined. First a general lower bound for the cost will be constructed and Theorems from other researches are stated which are relevant to the cost of the sensor network within this research. However, they need to be adapted and interpreted in the right way as they do not explicitly state a relation between the cost, the number of sensors and the probability on active sensors. Furthermore, with use of the theorems a new theorem will be stated and then the results of all the theorems will be compared to the simulations done with the grid and random uniform sensor network.

### 1 A general lower bound

In this section a new lower bound will be constructed. With this new lower bound a minimal cost can be calculated for arbitrary numbers of sensors and each value of  $p$ . The lower bound that will be constructed will be based on dividing the unit square surface into different squares. Then the will be looked at what happens when one of the squares does not contain any active sensors. This will result in a theorem about a lower bound for the cost for any sensor deployment. Formally the lower bound is described as follows.

**Theorem 4.1** (General lower bound). *Let  $n$  sensors be deployed on the square unit surface and let  $p$  be the probability on an active sensor, then for each sensor deployment  $x$ , and every  $\epsilon \in (0,1)$ , the cost of the sensor network  $C(x)$  satisfies, for sufficiently large  $n$*

$$C(x) \geq (1 - \epsilon) \frac{0.8}{\sqrt{2}} \sqrt{\frac{-\log(np)}{n \log(1-p)}}. \quad (4.1)$$

*Proof.* Divide the square unit surface into squares with sides  $\sqrt{2\alpha^2} \sqrt{\frac{\log(np)}{\pi pn}}$ , then the length from the middle of the square to a corner is equal to the half of the diagonal, that is:  $\alpha \sqrt{\frac{\log(np)}{np\pi}}$ . Then, the surface is divided into  $k = \frac{\pi pn}{2\alpha^2 \log(np)}$  (rounding errors are neglected) of such squares containing  $n_1, n_2, \dots, n_k$  sensors each. Now the cost is at least the length from the corner of a square to the middle of the square if there is at least one square which contains no active sensors. If such a square can be found with a probability of at least  $q$ , then we get that

$$C(x) \geq q\alpha \sqrt{\frac{\log(np)}{np\pi}}. \quad (4.2)$$

Now, such a  $q > 0$  will be specified. The probability of a square  $j$  having no active sensors is  $\bar{p}^{n_j} = (1-p)^{n_j}$ . Because the probability on active sensors is independent, the probability on all squares having some active sensors is equal to

$$Q = (1 - \bar{p}^{n_1})(1 - \bar{p}^{n_2}) \dots (1 - \bar{p}^{n_k}) \leq e^{-\bar{p}^{n_1}} e^{-\bar{p}^{n_2}} \dots e^{-\bar{p}^{n_k}}. \quad (4.3)$$

This must be bound above by  $1 - q$ . Now turning it into the logarithmic of  $Q$  and using an inequality of arithmetic and geometric means one obtains

$$-\log Q \geq \sum_{j=1}^k \bar{p}^{n_j} \geq k \left( \prod_{j=1}^k \bar{p}^{n_j} \right)^{1/k} = k(\bar{p})^{n/k}. \quad (4.4)$$

Using the value of  $k$  above, this results in

$$-\log Q \geq \frac{\pi n p (\bar{p})^n \frac{2\alpha^2 \log(np)}{\pi p^n}}{2\alpha^2 \log(np)} = \frac{\pi(np) \frac{2\alpha^2 \log(1-p) + 1}{\pi p}}{2\alpha^2 \log(np)}. \quad (4.5)$$

Name the right side of equation (4.5)  $c^*$ . Now  $\alpha$  can be chosen, such that the exponent of  $np$  is positive, then  $c^*$  is always positive and  $c^* \rightarrow \infty$  if  $n \rightarrow \infty$ :

$$\alpha^2 < \frac{-\pi p}{2 \log(1-p)}. \quad (4.6)$$

Then  $-\log Q \rightarrow \infty$  for  $n \rightarrow \infty$  and for each  $n$ ,  $-\log Q > 0$ . This means that  $Q$  is smaller than 1, and it can be bounded from above by  $1 - q$ . Since  $-\log Q \geq c^*$  is equal to  $Q \leq e^{-c^*}$  and  $1 - q \geq Q$ , it follows that  $q \leq 1 - e^{-c^*}$ . Now choose  $q = 1 - e^{-c^*}$  to attain the largest  $q$  to make the lower bound as large as possible. Then from equation (4.2) a lower bound is attained. Now for sufficient large  $n$ , choosing  $\alpha = 0.8 \sqrt{\frac{-(\pi p)}{2 \log(1-p)}}$  results in a large  $c^*$  such that  $q = 1 - e^{-c^*} \approx 1$ . Using  $q \in (0,1)$  the cost of a sensor network  $x$  can be bounded by below by

$$C(x) \geq q\alpha \sqrt{\frac{\log(np)}{np\pi}} = q \frac{0.8}{\sqrt{2}} \sqrt{\frac{-(\pi p)}{\log(1-p)}} \sqrt{\frac{\log(np)}{np\pi}} = q \frac{0.8}{\sqrt{2}} \sqrt{\frac{-\log(np)}{n \log(1-p)}}. \quad (4.7)$$

Because  $q \in (0,1)$ , it can be written as  $1 - \epsilon$ , where  $\epsilon \in (0,1)$  to indicate that it is close to 1 and to obtain (4.1).  $\square$

With this theorem a lower bound for the cost for general sensor networks is defined with the only assumption that  $n$  must be large enough. In the theorem a constant in  $\alpha$  is chosen as 0.8. The choice of this constant affects the lower bound as it changes the value of  $q$  and  $\alpha$ . The optimal constant can be found by maximizing  $q\alpha$  subjected to  $\alpha^2 < \frac{-\pi p}{2 \log(1-p)}$  and a closed form expression can be found containing the Lambert function. However this optimal solution does not add much improvement and therefore the derivation is not shown in this report.

## 2 Known theorems about sensor coverage

For stating more theorems about the cost of sensor networks, articles are used which focus on sensor coverage. Instead of seeking the cost or maximum radius needed for a certain sensor deployment to cover the whole area, a relation is required between the radius, number of sensors and the probability on failures which ensures coverage. In most cases the radius is held constant or is decreasing while the number of sensors is increasing. However the researches do not have the same objective, they can help understand the problem and similar methods may be applied to find relations which are useful in this research. For this research the most useful relations are

stated in an article about coverage in mostly sleeping sensor networks[6]. This paper is about finding necessary conditions which imply coverage for a sensor network with high probability for inactivity and the number of sensors going to infinity. In this paper important theorems are stated on the coverage conditions of the grid and random uniform deployments.

Now the theorems for the coverage conditions of the grid deployment will be given, however first some assumptions which are required to state the theorems. Throughout the paper it is assumed that an event  $T(x)$  almost always occurs if  $\lim_{x \rightarrow \infty} Pr[T(x)] = 1$ . Also, the radius  $r$ , probability on active sensors  $p$  and defined parameter  $c$  are functions of the number of sensors  $n$ . Assumed is that  $\limsup_{n \rightarrow \infty} p < 1$ ,  $np \rightarrow \infty$  as  $n \rightarrow \infty$  and  $r \rightarrow 0$  as  $n \rightarrow \infty$ . Besides that  $\phi(np)$  is a slowly growing function which monotonically increases and goes to infinity as  $n \rightarrow \infty$  and  $\phi(np) = o(\log \log np)$ . Lastly the parameter  $c$  is defined as

$$c(n) = \frac{np\pi r^2}{\log(np)}. \quad (4.8)$$

Now two theorems on coverage condition for the grid are stated, the proof can be found in the paper.

**Theorem 4.2** (Grid coverage condition). *If, for some slowly growing functions  $\phi(np)$ ,  $p$ , and  $r$  satisfy*

$$c(n) \geq 1 + \frac{\phi(np)(1 + \sqrt{p \log(np)}) + \log \log(np)}{\log(np)} \quad (4.9)$$

*for sufficiently large  $n$ , then the entire unit square region is almost always covered.*

**Theorem 4.3** (Grid non-coverage condition). *Assume  $\lim_{n \rightarrow \infty} p = 0$ . If, for some slowly growing function  $\phi(np)$ ,  $p$  and  $r$  satisfy*

$$c(n) = 1 - \frac{\phi(np)(1 + \sqrt{p \log(np)}) + \log \log(np)}{\log(np)} \quad (4.10)$$

*for sufficiently large  $n$ , then the entire unit square region is almost always not covered.*

For the random uniform deployment, similar conditions hold.

**Theorem 4.4** (Random uniform coverage condition). *Let  $n$  sensors be deployed uniformly over a unit square region. If, for some slowly growing function  $\phi(np)$ ,  $p$  and  $r$  satisfy*

$$c(n) \geq 1 + \frac{\phi(np) + \log \log(np)}{\log(np)} \quad (4.11)$$

*for sufficiently large  $n$ , then the entire unit square region is almost always covered.*

**Theorem 4.5** (Random uniform non-coverage condition). *Let  $n$  sensors be deployed uniformly over a unit square region, and assume  $\lim_{n \rightarrow \infty} pr^2 = 0$ . If, for some slowly growing function  $\phi(np)$ ,  $p$  and  $r$  satisfy*

$$c(n) = 1 - \frac{\phi(np) + \log \log(np)}{\log(np)} \quad (4.12)$$

*for sufficiently large  $n$ , then the entire unit square region is almost always not covered.*

So these 4 theorems are coverage conditions for the grid and random uniform deployment. This means that if these conditions hold, the square unit surface is almost always (not) covered and an explicit relation for the radius, the number of sensors and the probability on active sensors can be stated. However, the known theorems state coverage conditions instead of relations between the cost and the number of sensors and the probability on active sensors. Therefore they must be rewritten such that they can be compared with the simulations and the derived lower bound. This is done by using the value of  $c = \frac{np\pi r^2}{\log(np)}$  where  $r$  is considered the sensing radius of a sensor  $x$  and equations (4.9), (4.10), (4.11) and (4.12) can be rewritten as:

$$r \geq \sqrt{\frac{\log(np) + \phi(np)(1 + \sqrt{p \log(np)}) + \log \log(np)}{np\pi}} \quad (\text{grid, coverage}) \quad (4.13a)$$

$$r = \sqrt{\frac{\log(np) - \phi(np)(1 + \sqrt{p \log(np)}) - \log \log(np)}{np\pi}} \quad (\text{grid, no coverage}) \quad (4.13b)$$

$$r \geq \sqrt{\frac{\log(np) + \phi(np) + \log \log(np)}{np\pi}} \quad (\text{random uniform, coverage}) \quad (4.13c)$$

$$r = \sqrt{\frac{\log(np) - \phi(np) - \log \log(np)}{np\pi}} \quad (\text{random uniform, no coverage}) \quad (4.13d)$$

These equations describe the conditions for the cost of a sensor deployment for which the unit square surface is almost always (not) covered under the assumptions listed in this section. Therefore, it is expected that the cost of a grid sensor deployment is larger than  $r$  in equation (4.13b), but lesser or equal than  $r$  in equation (4.13a). For the random uniform deployment the cost  $C(x)$  is expected to be in between the values of  $r$  in equations (4.13d) and (4.13c). Now using the probability that the whole square unit surface is covered, a new theorem can be derived about the sensor cost.

**Theorem 4.6.** *Let  $n$  sensors be deployed uniformly over a unit square region, then under the assumptions of theorem 4.5 a lower bound for the costs can be stated as*

$$C(x) \geq r^- - o(r^-) \quad \text{as } n \rightarrow \infty, \quad (4.14)$$

where  $r^- = \sqrt{\frac{\log(np) - \phi(np) - \log \log(np)}{np\pi}}$  as in equation (4.13d) (where the unit square surface is almost always not covered).

*Proof.* Let  $J(x)$  be the cost of a random sensor deployment with a random set active sensors  $A$ . Then the probability that the area  $Q$  is covered by the sensor deployment  $x$  with the active set sensors  $A$  is equal to:  $\Pr(Q \text{ is covered}) = \Pr(J(x) \leq r)$ . Now the cost of a sensor deployment  $C(x)$  is the expectation of  $J(x)$  and is equal to

$$C(x) = \mathbb{E}(J(x)) = \int_0^{\sqrt{2}} 1 - \Pr(J(x) \leq r) dr = \int_0^{\sqrt{2}} \Pr(J(x) > r) dr. \quad (4.15)$$

Now the known theorems can be used to split up the integral. Let  $r^-$  be the  $r$  from the condition that the random uniform deployment is not covered (equation (4.13d)) and  $r^+$  the  $r$  from the condition that the random uniform deployment is covered (equation (4.13c)). Now the integral can be split up in 3 terms

$$\int_0^{\sqrt{2}} \Pr(J(x) > r) dr = \int_0^{r^-} \Pr(J(x) > r) dr + \int_{r^-}^{r^+} \Pr(J(x) > r) dr + \int_{r^+}^{\sqrt{2}} \Pr(J(x) > r) dr \quad (4.16)$$

Let  $E_+$  and  $E_-$  be the events that the unit square surface is covered or is not covered. Then if the actual sensing radius assigned to the sensors  $r \geq r^+$ , the whole unit surface is almost always

covered if  $n \rightarrow \infty$ , i.e.  $\lim_{n \rightarrow \infty} \Pr(E_+) = 1$ . Thus the minimal sensing radius, such that the area is covered,  $r = r^+$  should still result in  $\lim_{n \rightarrow \infty} \Pr(E_+) = 1$ . So the sensing radius required to cover the square unit surface is almost always smaller than  $r^+$ :  $\lim_{n \rightarrow \infty} \Pr(J(x) \leq r^+) = 1$ , so  $\lim_{n \rightarrow \infty} \Pr(J(x) > r^+) = 0$ . A similar argument can be used to a sensor network with a sensing radius  $r = r^-$  almost always does not cover the unit surface, i.e.  $\lim_{n \rightarrow \infty} \Pr(E_-) = 1$ . So with a sensing radius smaller than or equal to  $r^-$  the unit surface is almost always not covered, so the cost of sensor deployment is almost always larger than  $r^-$ :  $\lim_{n \rightarrow \infty} \Pr(J(x) \geq r^-) = 1$ . Now these results can be used to obtain a lower bound for the cost of a random uniform deployment which holds for large numbers of sensors. The discussion above implies that  $\Pr(J(x) > r^+) = 0 + o(1)$  and  $\Pr(J(x) \geq r^-) = 1 - o(1)$ . Using this result, one can obtain

$$\begin{aligned}
C(x) &= \int_0^{r^-} 1 - o(1) dr + \int_{r^-}^{r^+} \Pr(J(x) > r) dr + \int_{r^+}^{\sqrt{2}} 0 + o(1) dr \\
C(x) &= r^-(1 - o(1)) + (\sqrt{2} - r^+)o(1) + \int_{r^-}^{r^+} \Pr(J(x) > r) dr \\
C(x) &\geq r^-(1 - o(1)) = r^- - o(r^-).
\end{aligned} \tag{4.17}$$

□

So from the theorems stated in [6], a lower bound for the cost is derived which holds for large  $n$  and random uniform deployments. For the grid deployment, an assumption is made that  $p \rightarrow 0$  as  $n \rightarrow \infty$ , therefore similar statements cannot be made due to the assumptions of the theorems. For the random uniform deployment the assumption is made that  $pr^2 \rightarrow 0$  as  $n \rightarrow \infty$ , this is the case in this research as the cost of a sensor deployment  $C(x) \rightarrow 0$  as  $n \rightarrow \infty$  and can be interpreted as  $r \rightarrow 0$  as  $n \rightarrow \infty$ . In the simulations this lower bound will be simulated as  $r^-$  as it should give a good estimation and it is not known how large  $o(r^-)$  is.

### 3 Comparison with simulations

In this section the results from the known theorems from the paper will be compared with the derived general lower bound and the simulation data from chapter 3 of the random uniform deployment and the grid deployment. To do this, simulations of both the grid deployment and the random uniform deployment will be made and the costs of these simulations will be compared with the lower and upper bounds derived from the theorems, described by equation (4.13). Doing this, the performance and correctness of the theorems will be tested and more information will be provided. The theorems from the paper contain a variable  $\phi(np)$  which is a slowly growing function of the order  $o(\log \log(np))$ , which needs to be specified in order to do simulations. This function is used for the proof of the theorem and considered when  $n \rightarrow \infty$ . Therefore we can choose it as a small constant multiplied by  $\sqrt{\log \log(np)}$ , so the limit stays the same and the relation should hold. This way,  $\phi(np)$  is very small and the required radius for coverage in equations (4.13a) and (4.13c) are almost equal. Besides that, the radius which should result in no coverage in equations (4.13b) and (4.13d) are larger and almost equal as well. Since the constant can be chosen that small that it approximates 0, this will be used in the simulations in order to obtain only 1 lower and 1 upper bound from the paper.

Before looking at the simulation results, it is notable that without specifying the function  $\phi(np)$ , the equation (4.13c) for the grid deployment, is larger than equation (4.13d) for the random uniform deployment. This implies that the cost of the grid deployment is expected to be bigger than the cost of the random uniform deployment. From the simulations in chapter 3 it is known that this is not the case. The fact that the equation for the random uniform deployment is larger than the equation for the grid deployment may be explained by the assumptions. Because the relations

are stated for  $n \rightarrow \infty$  and so the asymptotic costs of both the deployments may be the same, or the random uniform can be even smaller. However, from the simulations this seems unlikely because in figure 3.4 the lines seem to be parallel.

In the figures 4.1 and 4.2 the simulations of the cost of the random uniform and grid deployment, the expected lower and upper bounds from the paper and the general lower bound can be seen. As concluded in chapter 3 the cost of the random uniform deployment is larger than the cost of the grid deployment. The black pluses and circles are values of the radius at which the unit square surface is almost always covered and not covered. So it is expected that the cost of both the random uniform and the grid deployment are always in between these two values. For the grid deployment this is the case for large  $p$ , however the random uniform deployment always has costs above these values. Therefore it can be concluded that the theorems from the paper cannot be used in this research as an upper bound for the cost. The reason for this is probably the number of sensors, which is going to infinity for the theorem stated in the paper and thus may not be large enough in the simulations as the error seems to get smaller when the number of sensors increases. The lowerbound derived from the paper does seem to hold, except for the probability of active sensors  $p$  close to 1 for the grid deployment. This may be caused by the assumptions on the grid deployment that  $p \rightarrow 0$  as  $n \rightarrow \infty$ , and as can be seen it does always hold for the random uniform deployment.

In the plots can be seen that the general lower bound holds for all the simulations, because it is lower than the cost for both the deployments. However there seems to be a big gap between the costs and the general lower bound, which is smaller for the lower bound derived from [6]. For small values of  $p$ , the general lower bound is slightly larger than the lower bound derived from [6]. Since many assumptions are made for the theorems in [6] and thereby the relations constructed for radius cannot be interpreted as a lower bound and upper bound for the grid deployment. The lower bound (4.14) for the random uniform deployment is better than the general lower bound, except for small  $p$ . In figure 4.2 it can be seen that when  $p$  is equal to 1, the cost of the grid deployments are lower than the lower bound (4.14). So this proves that the lower bound for the random uniform deployment does not hold for the grid deployment. The general lower bound is close to zero and thereby lower than the cost of the grid and thus holds for both the grid and random uniform deployments. Therefore the general lower bound should be used for the grid deployments. For the random uniform deployments, the lower bound from the paper should be used unless  $p$  is small.

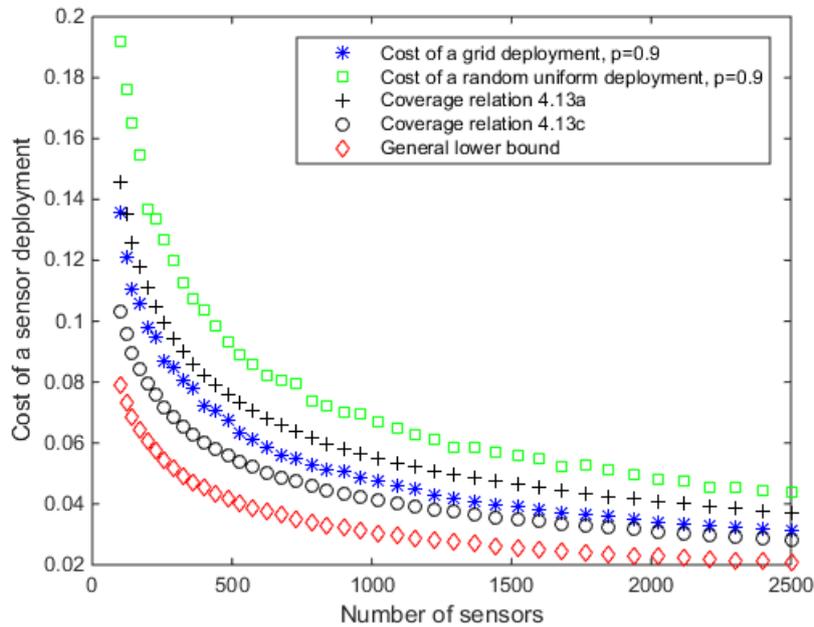


Figure 4.1: The simulated cost of the grid deployment and the random uniform deployment, the relations for the radius which implies coverage (4.13a) or no coverage (4.13c) and the general lower bound. The number of sensors  $n$  is varying from  $10^2$  to  $50^2$  and the probability on active sensors is kept equal at  $p = 0.9$ .

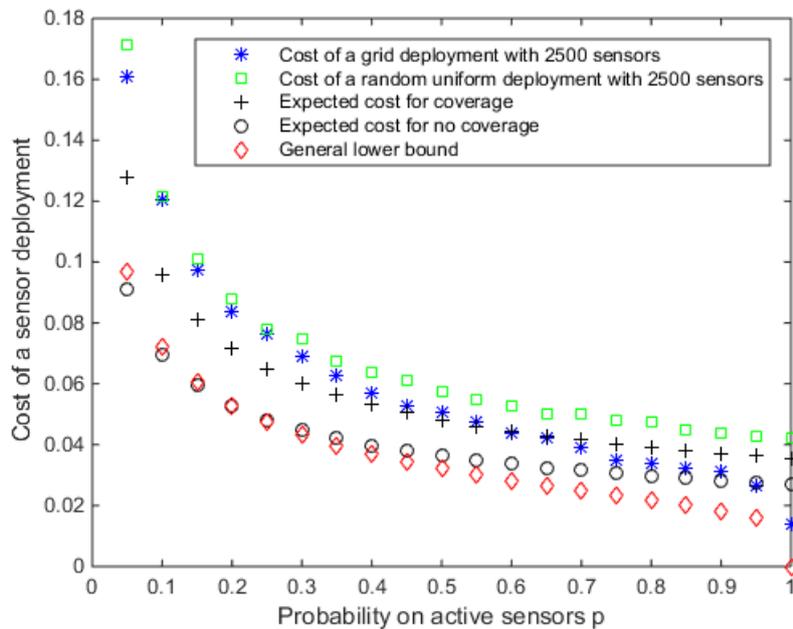


Figure 4.2: The simulated cost of the grid deployment and the random uniform deployment, the relations for the radius which implies coverage (4.13a) or no coverage (4.13c) and the general lower bound. The probability on active sensors  $p$  is varying from 0.05 to 1 and the number of sensors is kept equal at 2500.

## Chapter 5

# Sensors deployment algorithms

In this chapter a deployment strategy is described made by an algorithm which aims to minimize the cost of a sensor deployment. This algorithm uses the gradient to compute a descent direction in which the sensors will move. The algorithm is much alike the algorithm designed in [3], but now also takes failure rates into account. This algorithm will be described and analyzed on convergence. Then a second algorithm will be made which is derived from the first. This algorithm has quicker convergence in general, but will not be analyzed on convergence properties because it is much alike the first algorithm.

## 1 Gradient descent algorithm

The gradient descent algorithm starts with a random sensor deployment. Then at each iteration the cost is computed and the sensors will move in a direction which decreases the cost. In fact, the sensor at which the cost is attained in an set active sensors, will move towards the farthest point in its Voronoi partition. However, for each deployment there are many different possible sets active sensors and all of these will be taken into account. So the direction is based on a weighted sum of all farthest points in the Voronoi partitions for all possible sets active sensors. The set active sensors which is more likely to occur, will contribute more to the directions as it also does contribute more to the cost of the sensor deployment.

Now will follow the exact algorithm description. As stated, the algorithm is a gradient descent algorithm where at each iteration the location of the sensor changes based on the gradient of the cost function. The objective function, (2.3) has to be minimized with respect to the sensor locations. So again the average cost of a sensor network must be minimized:

$$C(x) = \sum_{A \neq \emptyset} Pr(E_A) \max_{i \in A} \max_{s \in V_i} \|s - x_i\|_2. \quad (5.1)$$

The algorithm is discrete and updates the sensor locations at each iteration. At each iteration  $k$  the sensors locations  $x_i^k$ ,  $i = \{1, 2, \dots, n\}$ , will be updated to  $x_i^{k+1}$  through the following equation:

$$x^{k+1} = x^k - \alpha \nabla C(x^k). \quad (5.2)$$

In this formula,  $\alpha$  is a positive constant smaller than 1 which should ensure convergence and the  $\nabla$  stands for the gradient with respect to  $x^k$ . Now we need to find a proper way to calculate the gradient,  $\nabla C(x^k)$ , at each step. To do this the cost function will be rewritten. For each iteration, let  $s_A^*$  be a point with maximal distance to the closest sensor for the active set of sensors  $A$ , so

$s \in W$ . And let  $x_A^*$  be closest active sensor to the point  $s_A^*$ . Assume that only 1 point be in  $W$  and that the sensor  $x_A^*$  is unique. Then the cost function can be formulated as

$$C(x) = \sum_{A \neq \emptyset} Pr(E_A) \|s_A^* - x_A^*\|_2. \quad (5.3)$$

Then the gradient can be expressed as follows:

$$\begin{aligned} \nabla C(x) &= \nabla_x \sum_{A \neq \emptyset} Pr(E_A) \|s_A^* - x_A^*\|_2 \\ &= \sum_{A \neq \emptyset} Pr(E_A) \nabla_x \|s_A^* - x_A^*\|_2. \end{aligned} \quad (5.4)$$

The distance is defined as the euclidean distance, as a consequence the derivative of the norm is as follows:

$$\nabla_x \|s - x\|_2 = \frac{x - s}{\|s - x\|_2}. \quad (5.5)$$

This equation describes the direction from the point  $s$  to the location of the sensor  $x$ , now by the minus sign from equation (5.2) it follows that the sensor will move in the direction of point  $s$ . From figure 5.1 it can be seen that for each active set of sensors, only 1 sensor will be move in the direction of the point at which the cost is attained. However, the direction in which a sensor will move is a weighted average over all the directions of all sets active sensors.

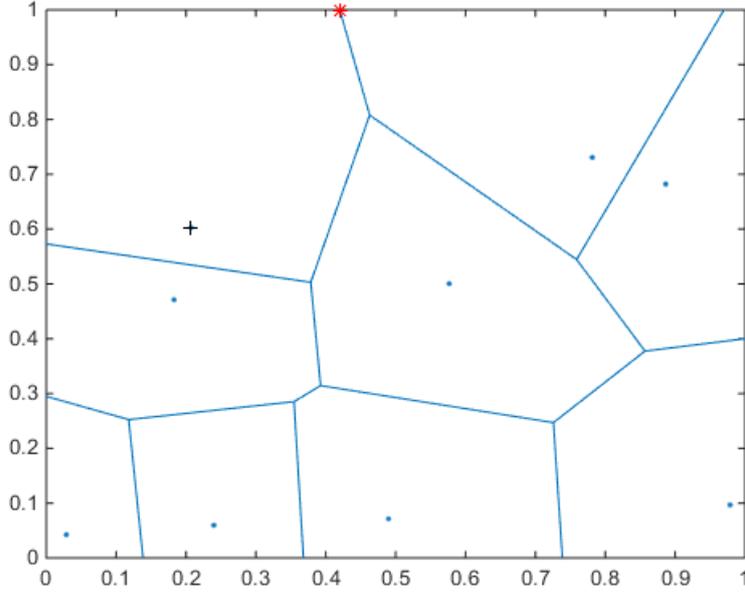


Figure 5.1: In this figure a sensor deployment with 9 active sensors can be seen together with the Voronoi diagram. The black plus is the only sensor that gains a direction with this active set of sensors, namely towards the red asterisk.

However, it is not necessary that the cost is attained at only 1 point by 1 sensor, therefore  $x_A^*$  and  $s_A^*$  do not have to be unique for all sets active sensors. To deal with the non uniqueness, the generalized gradient will be used which is defined as

$$\partial f(x) = \text{co}\left\{ \lim_{i \rightarrow \infty} df(x_i) \mid x_i \rightarrow x, x_i \notin S \cup \Omega_f \right\}. \quad (5.6)$$

In this equation  $co$  stands for the convex hull,  $\Omega_f$  is the set at which  $f$  is not differentiable and  $S$  denotes all sets with measure 0. In words the generalized gradient can be described as all possible convex combinations of all the possible derivatives which arise from the fact that the cost function  $C(x^k)$  has more derivatives when the cost is attained at more than 1 point or at more than one sensor. Now the generalized gradient can be used in the update function. Let  $g(x^k) \in \partial C(x^k)$ , then the update function can be rewritten as:

$$x^{k+1} = x^k - \alpha g(x^k) \quad (5.7)$$

Now instead of having one derivative which determines the direction the sensors will move to, all convex combinations of all possible derivatives can be used as a direction in which the sensor will move. This way the behavior of the solution of the algorithm does not have to be unique, this can be solved by taking the least-norm element of  $g(x^k) \in \partial C(x^k)$ . The properties of the gradient algorithm depend on the properties of the cost function  $C(x)$ , therefore it will be checked if the cost function  $C(x^k)$  is locally Lipschitz.

**Definition 5.1.** A function  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  is locally Lipschitz near  $x \in \mathbb{R}^N$  if there exist positive constants  $L_x$  and  $\epsilon$  such that  $|f(y) - f(z)| \leq L_x \|y - z\|$  for all  $y, z \in \mathbb{R}^N$  in the neighborhood of  $x$  such that  $\|x - y\| \leq \epsilon$  and  $\|x - z\| \leq \epsilon$ .

**Proposition 5.1.**  $C(x)$  is a locally Lipschitz for all  $x \in Q$ .

*Proof.* Let  $y_i \in Q, i = 1, 2, \dots, n$  be the locations of the sensors and let  $\epsilon_i$  be a  $1 \times n$  vector with small values such that  $z_i = y_i + \epsilon_i, z_i \in Q$ . Now an  $L_x$  will be specified and the function  $C(x)$  will be proven locally Lipschitz.

$$\begin{aligned} |C(z) - C(y)| &= \left| \sum_{A \neq \emptyset} Pr(E_A) \max_{i \in A} \max_{s \in V_i} \|s - z_i\|_2 - \sum_{A \neq \emptyset} Pr(E_A) \max_{i \in A} \max_{s \in V_i} \|s - y_i\|_2 \right| \\ &= \left| \sum_{A \neq \emptyset} Pr(E_A) (\max_{i \in A} \max_{s \in V_i} \|s - y_i - \epsilon_i\|_2 - \max_{i \in A} \max_{s \in V_i} \|s - y_i\|_2) \right|. \end{aligned} \quad (5.8)$$

Note that the Voronoi partitions within the summation are not the same because the Voronoi partitions depend on the locations of the sensors. Though a small change  $\epsilon_i$  in the location cannot result in a cost difference of more than  $\max_{i \in A} \|\epsilon_i\|_2$  for each set active sensors. Now will be explained why. Without loss of generality let  $A$  be the set active sensors and let  $C(z_A) \geq C(y_A)$ . Now assume the cost  $C(y_A)$  of the set active sensor is attained at a point  $s$ . Then this point  $s$  is either a vertex of the Voronoi partition ( $s \in W_1$ ), or a point on the boundary of  $Q$  ( $s \in W_2 \cup W_3$ ). Now if  $s$  is a point on the boundary of  $Q$ , the cost for this set active sensors can be increased at most  $\max_{i \in A} \|\epsilon_i\|_2$  if the closest active sensor moves a distance  $\max_{i \in A} \|\epsilon_i\|_2$  in the opposite direction of  $s$ . So for this case,  $C(z_A) - C(y_A) \leq \max_{i \in A} \|\epsilon_i\|_2$ . If  $s$  is a vertex of a Voronoi partition, then at least 3 sensors have the same distance to  $s$ . Therefore, moving only 1 of these sensors results in a different point with the largest distance to the corresponding sensor since the boundaries of the Voronoi partitions will be different. Therefore the maximal increase in distance is obtained if all the sensors with the point  $s$  in its Voronoi partition move in the opposite direction of  $s$ . Then the distance is again increase with at most  $\max_{i \in A} \|\epsilon_i\|_2$ , so also for this case,  $C(z_A) - C(y_A) \leq \max_{i \in A} \|\epsilon_i\|_2$ . Using this result, one can obtain:

$$|C(z) - C(y)| \leq \left| \sum_{A \neq \emptyset} Pr(E_A) \max_{i \in A} \|\epsilon_i\|_2 \right| \leq \max_{i \in [n]} \|\epsilon_i\|_2 \leq \|z - y\|_2. \quad (5.9)$$

□

Now it is known that the cost function is locally Lipschitz with Lipschitz constant equal to 1, the critical points can be defined as in [3].

**Proposition 5.2.** *Since  $C(x)$  is a locally Lipschitz function, if it attains a local minimum or maximum at  $x$ , then  $0 \in \partial C(x)$ , such a  $x$  is called a critical point.*

In the special case that the probability on active sensors is equal to  $p = 1$ , the critical points can be described as circumcenters of the Voronoi partitions. This means that the sensor locations are in the middle of their Voronoi partition. With a grid deployment, this is always the case. So the sensor locations of a grid deployment are critical points at which a local minimum is attained. This minimum is a local minimum and not a global minimum as shown in chapter 2. Now will be looked at the regularity of the cost function  $C(x)$ .

**Definition 5.2.** *A function  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  is regular at  $x \in \mathbb{R}^N$  if for all  $v \in \mathbb{R}^N$ , the right directional derivative of  $f$ :  $f'(x,v)$ , exists and equals generalized directional derivative of  $f$ :  $f^\circ(x,v)$ .*

Here the right directional derivative and the generalized directional derivative of  $f$  at  $x$  in the direction of  $v \in \mathbb{R}^N$  are defined as:

$$\begin{aligned} f'(x,v) &= \lim_{t \rightarrow 0^+} \frac{f(x+tv) - f(x)}{t} \\ f^\circ(x,v) &= \limsup_{t \rightarrow 0^+, y \rightarrow x} \frac{f(y+tv) - f(y)}{t}. \end{aligned} \quad (5.10)$$

It follows that locally Lipschitz functions which are convex or concave are also regular, as a result  $f(x) = \|x - s\|_2$  is regular for fixed  $s$ , since it is convex and locally Lipschitz. Now with help of the following proposition from [8] it can be proved that the cost function is regular.

**Proposition 5.3.** *Let  $\{f_k : \mathbb{R}^N \rightarrow \mathbb{R} | k \in \{1,2,3,\dots,m\}\}$  be a finite collection of locally Lipschitz functions near  $x \in \mathbb{R}^N$ . Consider  $f(x') = \max\{f_k(x') | k \in \{1,2,3,\dots,m\}\}$ . Then, if  $I(x')$  denotes the set of indexes  $k$  for which  $f_k(x') = f(x')$ , and if each  $f_i$  is regular at  $x$  for  $i \in I(x)$ , then  $f$  is regular at  $x$ .*

Now it can be shown that the cost function  $C(x)$  is regular for all  $x \in Q$ , a property that will be used to prove convergence.

**Proposition 5.4.** *The cost function  $C(x)$  from equation (5.1) is regular for all  $x \in Q$ .*

*Proof.* With the help of the geometrical properties of the cost function as stated in chapter 2, the maximal distance from a sensor  $x_i \in Q$  to the closest point  $s$  in its Voronoi partition can be rewritten. Therefore let  $W_1^i, W_2^i$  and  $W_3^i$  be the points of  $W_1, W_2$  and  $W_3$  which correspond to the sensor  $i$ .

$$\max_{s \in V_i} \|s - x_i\|_2 = \max \left\{ \max_{s \in W_1^i} \|s - x_i\|_2, \max_{s \in W_2^i} \|s - x_i\|_2, \max_{s \in W_3^i} \|s - x_i\|_2 \right\} \quad (5.11)$$

Then for each sensor, the maximal distance to the furthest point in the Voronoi partition is given as a maximum of a fixed finite set of locally Lipschitz and regular functions and therefore is regular by Proposition (5.3). Then the cost  $C(x_A)$  of a set active sensors can be written as:

$$C(x_A) = \max_i \left[ \max \left\{ \max_{s \in W_1^i} \|s - x_i\|_2, \max_{s \in W_2^i} \|s - x_i\|_2, \max_{s \in W_3^i} \|s - x_i\|_2 \right\} \right]. \quad (5.12)$$

This again is a maximum of a fixed finite set of locally Lipschitz and regular functions and therefore also regular by Proposition (5.3). Then the weighted sum over all possible active sets  $C(x_A)$ ,

which is equal to the cost function  $C(x)$  is a sum of regular functions and thereby also regular. So the cost function is regular for all  $x \in Q$ .  $\square$

Now the critical points are defined and some properties are derived, the convergence to the critical points must be examined. It is known that the negative gradient  $-\partial f(x)$  is always a descent direction if  $0 \notin \partial f(x)$ . Therefore La Salle's theorem will be used to prove convergence from the initial locations  $x^0$  to the critical points. However, before LaSalle's theorem can be used some more properties are required. Because the gradient is not continuous and can have more values if different sensors attain the cost in a set active sensors, the solutions to differential equation concerning the sensor locations,  $\dot{x} = f(x(t))$ , must be understood in the Filippov sense. A Filippov solution of  $\dot{x} = f(x(t))$  on an interval  $[t_0, t_1] \subset \mathbb{R}$  is defined as an absolute continuous solution of the differential inclusion

$$\dot{x} \in K[f](x),$$

where 
$$K[f](x) = \bigcap_{\delta > 0} \bigcap_{\mu(S)=0} \text{co}\{f(B_N(x, \delta) \setminus S)\}.$$
 (5.13)

In this formula,  $\mu$  denotes the Lebesgue measure. Now the Lie derivative of a function can be defined, this definition will be used to prove convergence. For a locally Lipschitz function  $g : \mathbb{R}^N \rightarrow \mathbb{R}$ , the set-valued Lie derivative of  $g$  with respect to  $f$  at  $x$  is defined as

$$L_f g(x) = \{a \in \mathbb{R} \mid \exists v \in K[f](x) \text{ such that } \zeta \cdot v = a, \forall \zeta \in \partial g(x)\}. \quad (5.14)$$

Note that for each  $x$ , the Lie derivative equals a closed and bounded, possible empty interval. The following theorem is a nonsmooth version of LaSalle's invariance principle [9].

**Theorem 5.1** (LaSalle's invariance principle). *Let  $g$  be a locally Lipschitz and regular function. Then for  $x_0 \in \mathbb{R}$ , let  $g^{-1}(\leq g(x_0), x_0)$  be the connected component of  $\{x \in \mathbb{R}^N \mid g(x) \leq g(x_0)\}$  containing  $x_0$ . Then assume that the set  $g^{-1}(\leq g(x_0), x_0)$  is bounded and either  $\max L_f g(x) \leq 0$  or  $L_f g(x) = \emptyset$  for all  $x \in g^{-1}(\leq g(x_0), x_0)$ . Then  $g^{-1}(\leq g(x_0), x_0)$  is strongly invariant for  $\dot{x} = f(x(t))$ . Let  $Z_{f,g} = \{x \in \mathbb{R}^N \mid 0 \in L_f g(x)\}$ , then any solution  $x$  starting at  $x_0$  converges to the largest weakly invariant set  $M$  contained in  $Z_{f,g} \cap g^{-1}(\leq g(x_0), x_0)$ .*

With use of this theorem it can be proved that the sensors locations converge to the critical points. Instead of the discretized system, the continuous system will be used to prove the convergence. From the results from the continuous system we can conclude that with proper choice of the constant  $\alpha$  the discrete system has the same convergence properties. Let the cost function be defined as  $C(x)$  from equation (5.1) and the movement of the sensors  $x$  be determined by

$$\dot{x}(t) = -\partial C(x(t)). \quad (5.15)$$

Then  $Q$  is a bounded area in which the sensors  $x$  are, with initial sensor locations  $x_0$ . Now because the area to which the sensors belong is bounded and the cost is always larger than zero and lesser than  $\sqrt{2}$ , the values of the cost function  $C(x)$  are always bounded and so are the trajectories of the sensors  $C^{-1}(\leq C(x_0), x_0)$ .

**Theorem 5.2.** *Let  $x_0 \in Q$  be the locations of the sensor and let  $C^{-1}(\leq C(x_0), x_0)$  be the bounded trajectories of the sensors. Then any solution  $x : [t_0, \infty) \rightarrow \mathbb{R}^N$  of (5.15) starting at  $x_0$  converges asymptotically to the set of critical points of  $C(x)$  contained in  $C^{-1}(\leq C(x_0), x_0)$ .*

*Proof.* Let  $a \in L_{-\partial C(x(t))} C(x)$ , then by the definition of the Lie derivative, there exists a  $w \in K[-\partial C](x) = \partial C(x)$  such that  $a = \zeta \cdot w$  for all  $\zeta \in \partial C(x)$ . Now the particular choice  $\zeta = -w \in \partial C(x)$  gives  $a = -\|w\|^2 \leq 0$ . Therefore,  $\max L_{-\partial C(x(t))} C(x) \leq 0$  or  $L_{-\partial C(x(t))} C(x) = \emptyset$ . It is shown already that the function  $C(x)$  is locally Lipschitz and regular, therefore LaSalle's invariance principle can be used to deduce that any solution  $x : [t_0, \infty) \rightarrow \mathbb{R}^N$  starting at  $x_0$  converges

asymptotically to the largest weakly invariant set  $M$  contained in  $\bar{Z}_{f,g} \cap C^{-1}(\leq C(x_0), x_0)$ . This in fact are critical points where  $0 \in \partial C(x)$ .  $\square$

So now it is proven that the continuous system converges for all possible initial locations to the set of critical points. From there we can deduce, that if the continuous system is discretized as in (5.7), the sensors locations converge to the set of critical points when the scalar  $\alpha$  is chosen properly.

**Remark 5.1** (coincident sensors). *In the simulations, but also in the continuous case it may happen that sensors fall together on the same location. At that point, the incident sensors have equal Voronoi Partitions and therefore will be merged. Then the direction of the merged sensor is the same as the two original sensors and the probability on being active must be adjusted to  $1 - (1 - p)^2$ . The event that two sensors have incident locations may have 2 reasons, namely the scalar  $\alpha$  may be too large, or the sensors converge to the same location. The scalar  $\alpha$  only affects the discrete case where the sensors move in specific directions such that they fall exactly together. This is a bad choice of  $\alpha$  which must be avoided and shows that it should have been chosen smaller. For values of  $p$  very close to 0, the sensors will all move too the middle and they can converge to the same location:  $[0.5, 0.5]$ . This is not a problem, but the sensors converge in this case to critical points. When two sensors converge to the same locations, the regularity of  $C(x)$  is not affected and the function remains locally Lipschitz.*

Simulations are done in order to see the actual results of the gradient algorithm. For these simulations, the initial sensor locations  $x^0$  are drawn from a random uniform distribution over  $[0, 1] \times [0, 1]$ . The cost is computed as described in chapter 2 by considering all points of  $W_1$ ,  $W_2$  and  $W_3$  and compare which one has the largest distance to the closest sensor. The constant  $\alpha$  influences the convergence of the algorithm, if it is chosen very small the algorithm will certainly converge, but it may take a lot of steps  $k$  and thereby too much time. In these simulations  $\alpha$  is chosen equal to  $\frac{1}{500}$  which gives good results, but does not take too much time. Because at each step  $k$  the cost for all possible sets active sensors is computed to assign new locations for the sensors at step  $k + 1$ , the algorithm is rather slow and the number of sensors exponentially increases the computational time. Besides that only 1 sensor gains a new direction for each active set sensors and as a result the convergence is slow. Therefore, the number of sensors cannot be very large and is chosen as  $n = 9$  as the cost of the deployment can be compared to a grid deployment.

In chapter 2 it is already shown that the grid deployment is not optimal for  $p = 1$  and the algorithm can make a deployment strategy with lower cost. Then the constructed deployment has Voronoi partitions which seem more alike hexagons instead of the squares from the grid deployment. In figure 5.2 can be seen what happens when the algorithm is applied to an initial sensor deployment  $x^0$  with 9 sensors and  $p = 0.9$ . The cost seems to be converging to a local minimum and the grid deployment has a higher cost, as can be seen in figure 5.3.

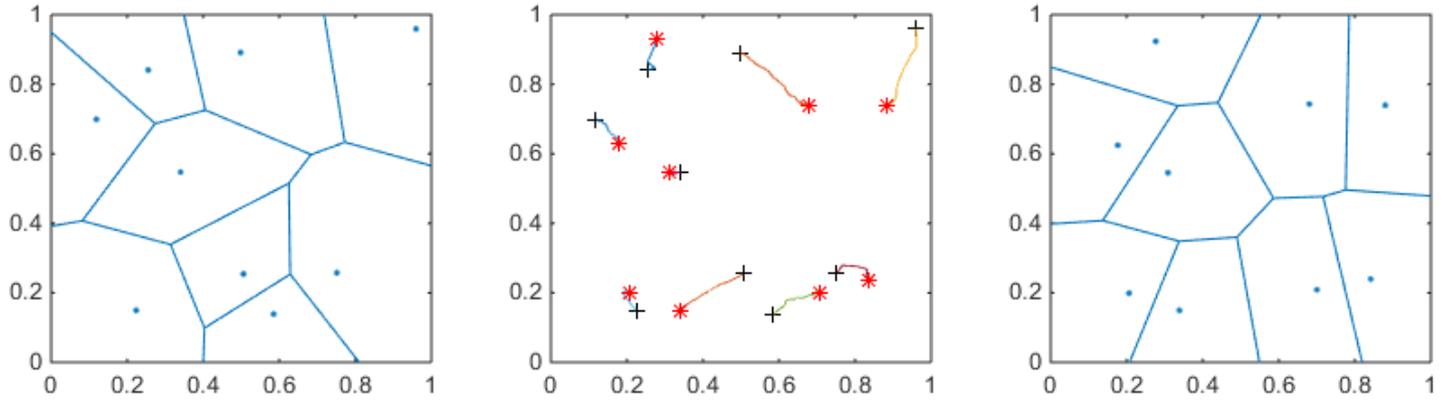


Figure 5.2: A simulation outcome from the gradient descent algorithm with 9 sensors. At the left side the initial sensor locations  $x^0$ , in the middle the paths of the sensors throughout all 5000 steps and at the right the final sensor locations.

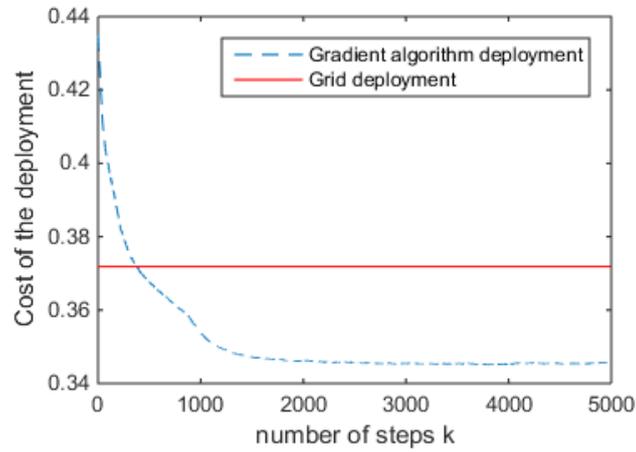


Figure 5.3: The cost of the deployment at the steps  $k$  in the algorithm together with the cost of the grid deployment.

## 2 Gradient based algorithm

In this section an algorithm will be described which is very similar to the gradient algorithm, but should have faster convergence. Because the aim to minimize the cost is the same as with the gradient algorithm, the objective function  $C(x^k)$  remains the same and is as in equation (5.3). Also the update step is similar to (5.7), but now the gradient will not be used. In this algorithm, each set active sensors will contribute to the directions in which all sensors in the set will move. This means that not only the sensor which attains the cost will gain direction, but also the other sensors will. Let  $h(x^k)$  be the direction in which the sensors will move, than update function is defined as:

$$x^{k+1} = x^k + \alpha h(x^k). \quad (5.16)$$

Here  $\alpha$  is a constant which should ensure convergence. The direction  $h(x_i^k)$  of a sensor  $i$  is chosen as the weighted average over all directions towards the farthest points in the Voronoi partitions  $V_i$  of all possible sets of active sensors in which  $i$  is contained. Now it may happen that more points in the Voronoi partition attain the maximal distance to the sensor, in that case a convex combination is allowed as direction. Now the direction of a sensor  $i$ ,  $h(x_i^k)$  can be written as:

$$h(x_i^k) \in H(x_i^k) = \sum_{A \neq \emptyset, i \in A} Pr(E_A) \text{co}\{s - x_i^k \mid \|s - x_i^k\|_2 \geq \|t - x_i^k\|_2, \forall t \in V_i, s \in V_i\}, \forall i \in [n] \quad (5.17)$$

In the figures 5.4 and 5.5 a simulation with this gradient based algorithm can be seen. The result of this algorithm is very similar to the results shown in 5.2 and 5.3, however the convergence is a bit faster. The faster convergence is due to the fact that at each step active set of sensors the sensors are moving towards the circumcenter of their Voronoi partition, which is in general the best way to lower the cost. In general the simulation has much quicker convergence, especially when the sensors start clustered in a corner. Besides for  $p = 1$ , the algorithm works much faster as it does move all the sensors, while the gradient descent algorithm only moves 1 sensor at each step.

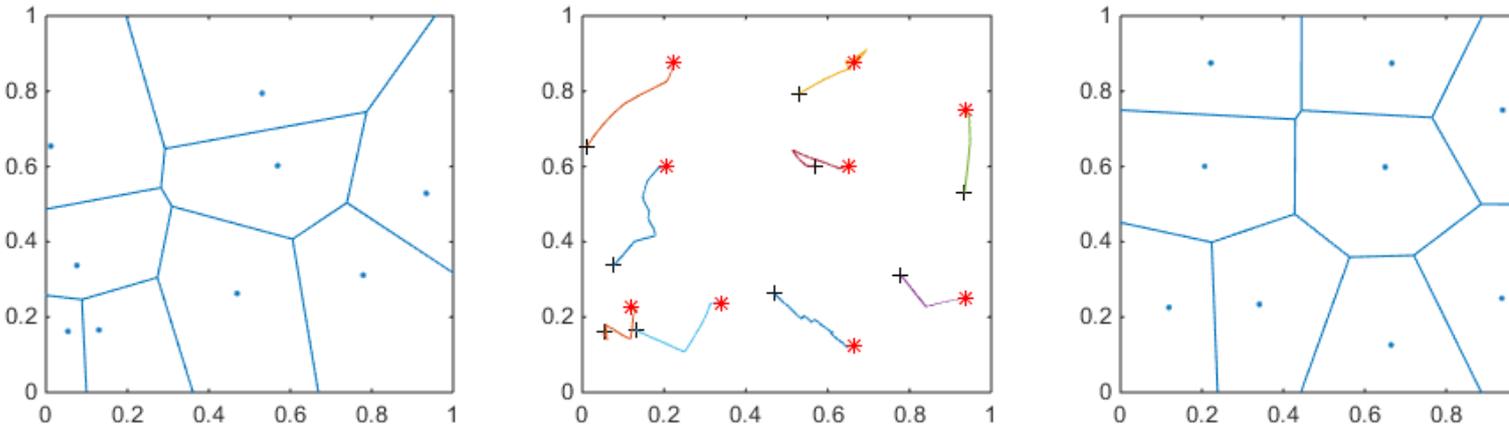


Figure 5.4: A simulation of a sensor deployment with 9 sensors made with the algorithm based on the gradient descent algorithm. At the left side the initial sensor locations  $x^0$ , in the middle the paths of the sensors throughout all 5000 steps and at the right the final sensor locations.

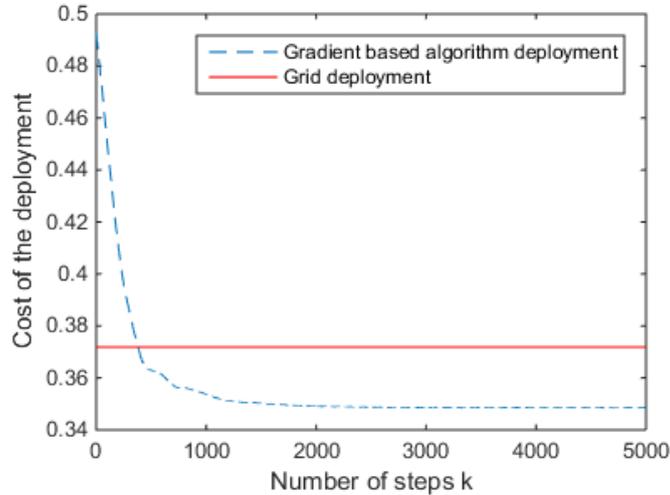


Figure 5.5: *The cost of the deployment at the steps  $k$  in the algorithm together with the cost of the grid deployment.*

Because the algorithm is very similar to the gradient algorithm, the critical points and the convergence is not analyzed. The set of critical points is not exactly the same set, but is closely related. To make a good comparison between the two algorithms, simulations have been done for 5 different initial sensor locations (9 sensors), sampled according to the random uniform deployment. In figure 5.6 on the next page the costs for the two algorithms can be seen for 5 different initial sensor locations. As can be seen, 4 out of 5 times the gradient based algorithm is decreasing faster than the gradient algorithm, which suggests that the non gradient algorithm has faster convergence in most cases. For the gradient algorithm 5000 update steps is not always enough for convergence and the cost still seems to be decreasing for some initial conditions. For the non gradient algorithm, the cost seems to be stabilized below the cost of the grid deployment for all different initial conditions.

Besides that it can be seen that the gradient algorithm always has a decreasing cost, while the gradient based algorithm has small drawbacks. This brings up the question whether the directions  $h(x_i^k)$  are always descending directions or maybe they are mostly descending directions which may not always result in decreasing cost. But it can also be the result of the discretization and maybe the scalar  $\alpha$  used in the update step is too big. This may push the sensors in a different trajectory to different or even the same critical points which results in a small drawback in the cost. For the gradient descent algorithm, this does not seem to be a problem and the cost is always decreasing and in the most plots it still is decreasing and may attain a lower cost than the non gradient. Among all 5 simulations with different initial conditions, the lowest cost is attained by the gradient algorithm.

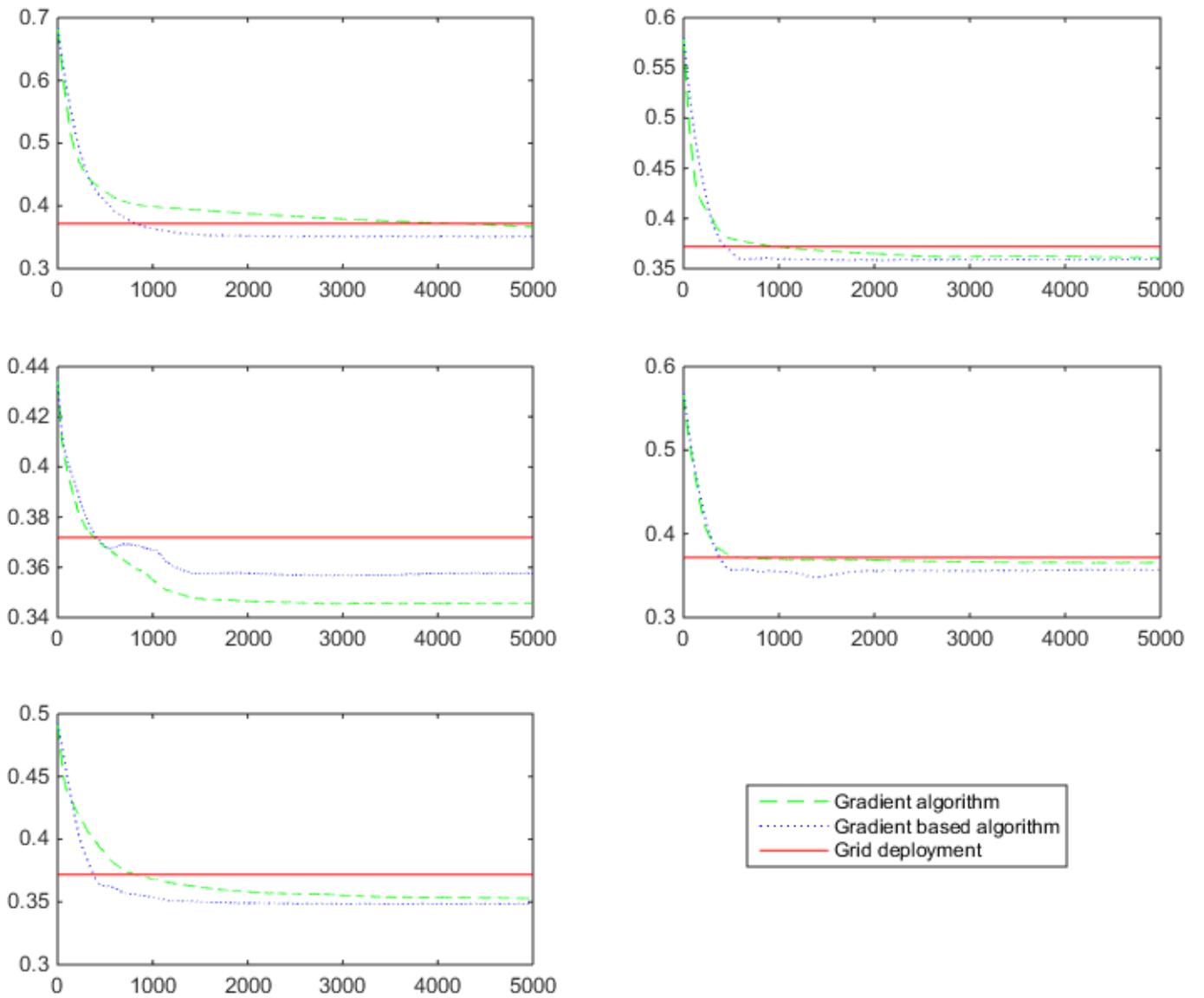


Figure 5.6: *The cost of the deployments made by the gradient and gradient based algorithms at different steps  $k$  together with the cost of the grid deployment. Each different initial condition is in a different plot for a one on one comparison.*

# Chapter 6

## Discussion

In this chapter the results of this research are discussed. The theorems and performance of the gradient and non gradient algorithm will be discussed here. This will be done starting at the first chapter and then working down to the last chapter about the algorithms.

First of all the cost function which expresses the quality of a sensor network by looking at the average cost. This because the number of failures is probabilistic and therefore not known, so an average should give a good estimation of the expected cost. However, the cost of an active set of sensors is chosen as the worst case measurement which may not be as logical. The maximal distance from any point to the closest sensor may be an extreme way to measure the quality and instead the quality could be expressed as an integral over the whole domain of a function depending on the distance of each point to the closest sensor. This might result in a more stable network but as can be read in [2] it causes the sensor to tend to be more in the middle and the points near the boundary are less important. However the choice for the cost as a the maximal distance to the closest sensor in some cases corresponds to the disk-covering problem of which much more is known [1].

In chapter 2 the grid and the random uniform deployment strategies are discussed. Both these strategies are natural but suboptimal heuristic deployment strategies. In the one-dimensional case the performance of the equispaced sensor network was optimal for the case that sensors may not fail. Therefore it was expected that the grid would perform optimal as well without failures. From the simulations with the gradient it is known that the grid is not optimal. This itself is a good result, however the optimal solution still cannot be described explicitly by any deployment, even without failures. However the structure of the optimal solution without failures seem to have a hexagon Voronoi structure. In [3] an algorithm is defined to find the locations of the sensors which give local minimal cost. In that research it is proven that the problem is equal to finding sensor locations which are the circumcenters of their own Voronoi partition. Thereby, it can be concluded that the optimal solution is a deployment where the sensor locations are circumcentres of their Voronoi partition, however there will remain more possible solutions to this problem and an optimal solution cannot be described.

From a paper on sensor coverage with sleeping sensors ([6]) coverage conditions are obtained which depend on the sensing radius of a sensor deployment. Expected was that the value of the costs of the grid and the random uniform deployments would be in between the relations for coverage and non-coverage, however for most cases this was not true. This can be related to the many assumptions which were made such as the number of sensors going to infinity. In the simulations this is of course not possible and the number of sensors may have been too small. However the theorem for coverage for the random uniform deployment has a lower requirement for the radius

to cover the whole area, than for the grid deployment as can be seen in (4.13a) and (4.13c). This arises the expectation that the radius or cost of the uniform deployment should be smaller than that of the grid deployment. This is in none of the simulations the case, however the failure rate is assumed very small in the theorem and the number of sensors is very large. This might be the reason for the lower requirement and it might be the case that for low chances on active sensors, the asymptotic cost of the random uniform grid is lower. Besides that the general lower bound which is constructed in this paper, still yields some assumptions and thereby may not hold for small number of sensors and small probabilities on active sensors. This is also a bit dissatisfying, because if the theorems will be implemented, it is probably not very common that a very large number of sensors are deployed.

The algorithms defined in chapter 5 are algorithms which produce good results, however there are some disadvantages. The number of sensors in the simulations has to be very small in order to make it computable in a reasonable time. Therefore the results cannot be compared to the theorems about the lower bounds in chapter 4 which use larger number sensors, but instead complement them. Besides that the initial condition and the scalar used in the update step do affect the convergence in both the final locations and the rate of convergence. Especially the final solution would give more information if it was always the global optimum, but global optimal are also not found in [3] where the sensors cannot fail at all.

## Chapter 7

# Conclusion

In this research the cost of sensor deployments are analyzed on several things. Firstly the cost a sensor as defined in this research has only 3 sorts of points which determine the actual cost of an active set of sensors. The cost is defined by the worst measurable point in the area, because of that the problem of finding the lowest cost with no failing sensors is very similar to the disk-covering problems. However, this measure of the cost of a sensor deployment did not allow a linear program such as in the one-dimensional case in which the optimal cost and sensor placement could be computed. To analyze the behavior of the cost when more sensors are used or different probabilities on active sensors, a grid deployment and a random uniform deployment were introduced. From the one-dimensional case it was expected that the grid is optimal with no inactive sensors, it can be shown that this is not the case in the two-dimensional setting. Also, from the simulations it is expected that the both the deployments have a cost which depend logarithmic on the number of sensors.

Instead of finding the optimal cost, creating lower and upper bounds for the cost will also provide information about the cost behavior. Therefore theorems of a paper which state coverage conditions were rewritten to a relation about the radius. From these conditions a lower bound is made for the random uniform deployment for large numbers of sensors, but due to the assumptions, no other lower or upper bounds could be made. Besides that, a more general lower bound was constructed for the cost which also holds for a large enough number of sensors. This general lower bound is under most circumstances less strict but can be used in more settings.

Lastly two algorithms are defined which aim to minimize the cost of a sensor deployment. While both of the algorithms converge to local optimal solutions, the global optimal solution still is unknown. The first algorithm uses the gradient direction to update the sensor locations and the second algorithm uses a gradient based direction which has faster convergence.

## Chapter 8

# Recommendations

In this chapter, recommendations will be done for further research. Due to the time limit some things are not done in this research, besides that, choices were made which influences the research and with different choices, different results can be obtained.

The first thing on which more research can be done is a different cost function. In this research, the cost function is defined as a weighted average over all possible sets active sensors and then the cost for an active set sensors is the farthest distance from a point to the closest sensor. Instead of looking at the worst measurable point, one can look at all points and take the average over the whole field. This can be done by replacing the maximal value by an integral over the whole area. Then the boundary of the area on which the sensors are placed may have a smaller influence, especially for small number of sensors.

Instead of changing the cost function, the area itself could be changed as well. In some researches the area is already varied as a polygon and as long as the area is convex, the model should still be applicable with some minor adjustments. Though it can help arise more understanding for the optimal cost and the cost behavior for different probabilities on active sensors.

Probably the most interesting is finding and optimal solution for the problem of robust sensor coverage. As a start of finding the optimal solution, an optimal solution can be searched for in the case that the probability on active sensors is equal to 1. In this research it is already shown that the grid deployment is not optimal, but instead a honeycomb alike structure seems to have better results and may even be optimal. Then the task is to find how the honeycomb should be defined and how it changes, when the area which must be covered changes. Then if an optimal deployment is found with no failing sensors, it may be used to derive a general solution for when failing sensors are allowed. Besides that it may be of use to make a better and more general lower bound for the cost of the sensor network.

For the gradient descent algorithm as defined in this research, the convergence to critical points is shown. However the algorithm can have very slow convergence. Besides that, the critical points are local minima and thus may not be the optimal solutions. To improve the convergence, different algorithms can be used, such as the adapted version of the gradient algorithm or a greedy algorithm. A greedy algorithm for a sensor deployment is an algorithm which chooses locations for the sensors one after another taking the failure rate into account. The greedy algorithm could result in a very fast algorithm, however the results are often not that good. Therefore, the algorithms can be combined to have the best result and the fastest convergence.

# Bibliography

- [1] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.
- [2] S. Hutchinson and T. Bretl, “Robust optimal deployment of mobile sensor networks,” pp. 671–676, 2012.
- [3] J. Cortés and F. Bullo, “Coordination and geometric optimization via distributed dynamical systems,” *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.
- [4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” in *Robotics and Automation, 2002. Proceedings. ICRA’02. IEEE International Conference on*, vol. 2, pp. 1327–1332, IEEE, 2002.
- [5] S. Shakkottai, R. Srikant, and N. B. Shroff, “Unreliable sensor grids: Coverage, connectivity and diameter,” *Ad Hoc Networks*, vol. 3, no. 6, pp. 702–716, 2005.
- [6] S. Kumar, T. H. Lai, and J. Balogh, “On  $k$ -coverage in a mostly sleeping sensor network,” pp. 144–158, 2004.
- [7] P. Frasca, F. Garin, B. Gerencser, and J. M. Hendrickx, “One-dimensional coverage by unreliable sensors,” *arXiv preprint arXiv:1404.7711*, 2014.
- [8] F. H. Clarke, *Optimization and nonsmooth analysis*. Siam, 1990.
- [9] A. Bacciotti and F. Ceragioli, “Stability and stabilization of discontinuous systems and nonsmooth lyapunov functions,” *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 4, pp. 361–376, 1999.